Universidade Nova de Lisboa

Faculdade de Ciências e Tecnologia

Departamento de Informática

Dissertação de Mestrado

SOLAP+

Ricardo Filipe Silva

Orientador: Prof. Doutor João Moura Pires

Trabalho apresentado no âmbito do Mestrado em Engenharia Informática, como requisito parcial para obtenção do grau de Mestre em Engenharia Informática.

Julho 2010

Resumo

No início do século XXI, Bédard propôs incorporar ao modelo multi-dimensional dados geográficos, originando o conceito SOLAP. Através deste conceito, os analistas puderam obter melhores análises das estruturas e relações de dados espaciais, mantendo as características benéficas que provêm dos sistemas OLAP, isto é, informação sumarizada, análise de dados a diferentes níveis de granularidade, exploração interactiva dos dados, etc.

Devido à adequação dos sistemas SOLAP, face aos sistemas OLAP para o processo de suporte à decisão, algumas aplicações têm sido desenvolvidas. Porém, estas aplicações têm sido concebidas para um contexto específico. Com o propósito de se transpor esta limitação, vários foram os trabalhos que culminaram num modelo genérico SOLAP.

Apesar de este modelo solucionar diversas limitações de aplicações anteriores, actualmente, não suporta análises com duas entidades espaciais em simultâneo, por exemplo "total de voos entre o aeroporto *X* para o aeroporto *Y*" ou "qual a quantidade de resíduos lançados por uma indústria *X* no rio *Y*?". Por outro lado, a visualização dos mapas pode despoletar o efeito contrário ao desejado. Facilmente o mapa poderá ficar desorganizado devido ao excesso de objectos geográficos presentes, o que prejudica a visualização/análise de dados espaciais.

Para dar resposta a estas questões, esta dissertação pretende estender o modelo SOLAP genérico, de modo a suportar análises onde estão presentes duas entidades espaciais em simultâneo e integrar algoritmos de agrupamento espacial, com o objectivo de garantir a visibilidade do mapa em situações de excesso de objectos geográficos.

Abstract

In the early 21st century, Bédard proposed incorporate spatial data to the multi-dimensional model, originating the concept SOLAP. Through this concept, analysts were able to obtain better analysis of the structures and relations of spatial data, retaining the beneficial characteristics that come from OLAP systems, i.e., summarized information, analysis of data in different levels of granularity, interactive exploration of data, etc.

Due to the adequacy of SOLAP systems, compared to OLAP systems, for the process of decision support, some applications have been developed. However, these applications have been designed for a specific context. With the purpose to overcome this restriction, there were several works that culminated in a generic SOLAP model.

Although this model solve several limitations of previous applications, currently do not support analysis with two spatial entities simultaneously, for example "total flights between the airport X to airport Y" or "what is the amount of waste released by an industry X in the river Y?". On the other hand, maps visualization can trigger the opposite effect. Easily the map may become disorganized due to the excessive spatial objects present, which affect the visualization / analysis of spatial data.

To answer these issues, this thesis intends to extend the generic SOLAP model, in order to withstand analysis where there are two spatial entities simultaneously and integrate spatial clustering algorithms, in order to ensure the visibility of the map in situations of excessive spatial objects.

Agradecimentos

Os meus agradecimentos vão para aqueles que tornaram, de alguma forma, possível a realização desta dissertação.

Para o Professor Doutor João Moura Pires por ter, em primeiro lugar, apostado em mim para dar continuidade a uma linha de trabalho. Segundo, agradeço pela sua orientação e por todas as discussões interessantes e conselhos que me colocaram na direcção correcta.

Um agradecimento especial para a Carla Diogo, não só pela ajuda que deu na revisão do documento, mas também porque é uma pessoa importante para mim.

Para a empresa ViaTecla, pela sua colaboração, facultando um conjunto de dados que permitisse realizar um dos casos de estudo desta dissertação.

Para todos os amigos que, de um modo ou outro, me ajudaram, em especial para aqueles que me acompanharam durante o percurso académico: Fábio Neves, David Sacramento, Nuno Alves, Nuno Parreira, Miguel Domingues.

Finalmente, um agradecimento aos meus pais por tudo aquilo que me souberam transmitir.

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Este capítulo apresenta o contexto, a motivação e o objectivo desta dissertação. No final é apresentada a sua estrutura.

Este capítulo apresenta o contexto e a motivação para o desenvolvimento desta dissertação, apresentando alguns conceitos introdutórios sobre o modelo multi-dimensional. Menciona os objectivos e contribuições realizadas e apresenta uma visão global da sua estrutura.

1.1 Contexto e Motivação

Ao longo dos últimos anos, muitas foram as questões colocadas aos sistemas OLAP. *Quais os dez produtos mais vendidos? Quando é que são vendidos? Quais são as lojas que vendem mais? Quais os factores demográficos que afectam as vendas? Onde é que se encontram as lojas mais lucrativas?*

Em geral, as questões colocadas pelos utilizadores OLAP inserem-se nas categorias *quem?*, *o que?*, *porquê?*, *quando?* e *onde?*. Apesar destas ferramentas realizarem um óptimo trabalho ao apresentar os dados sumarizados para as diversas análises, para dar resposta às questões *onde?* a maneira mais intuitiva é através do recurso a mapas [**2**].

A maioria das aplicações OLAP focam-se em dados textuais e em métricas numéricas, embora estudos concluam que 80% dos dados se refiram a informação espacial [**3**]. Como é evidente, esta quantidade significativa de dados, aliada à necessidade de melhorar as análises associadas ao *onde?*, tornou adequada a integração da informação geográfica dentro do processo de tomada de decisão, originando o conceito de *spatial* OLAP (SOLAP).

SOLAP é definido por Bédard como "*a visual platform built especially to support rapid and easy spatio-temporal analysis and exploration of data following a multidimensional approach comprised of aggregation levels available in cartographic displays as well as in tabular and diagram displays*" [**4**]. Com este conceito, a visualização e a exploração de dados espaciais foram adicionadas aos típicos sistemas *OLAP*, que por sua vez têm por base uma estrutura multi-dimensional.

A utilização do modelo dimensional advém de variados propósitos [**5**]. A simplicidade deste modelo facilita a tarefa dos analistas, permite interrogações eficientes e constitui a base para o desenvolvimento de *software* genérico, que proporciona aos utilizadores navegar em grandes conjuntos de dados de um modo intuitivo.

No modelo multi-dimensional encontram-se os seguintes conceitos chave:

- Hiper-cubo (Tabela de facto);
- Métricas;
- Dimensões;
- Atributos;
- Níveis;
- Hierarquias;
- Granularidade;
- Facto.

O ***hiper-cubo*** é uma estrutura lógica que contém os dados indexados por dimensões, associado a um conjunto de métricas (ex: quantidade vendida, preço de custo).

As ***dimensões*** são utilizadas para mapear e categorizar os dados em entidades relevantes para uma qualquer organização (ex: produto, cliente, loja). Os ***atributos*** caracterizam a respectiva entidade (ex: nome, morada, etc.).

Cada dimensão pode estar organizada segundo diferentes ***níveis*** de granularidade. As ***hierarquias*** são definidas utilizando os diferentes níveis da dimensão. Por exemplo, a dimensão *cliente* pode ter a seguinte hierarquia: freguesia, concelho, distrito, país e continente.

Na ***tabela de facto*** encontram-se os dados ao nível mais fino de granularidade, definido no respectivo modelo. Sempre que se queira agregar os dados são utilizados os níveis superiores das dimensões.

A ***granularidade*** dos dados refere-se ao nível de detalhe com que os dados são guardados nas tabelas de facto. O termo ***facto*** representa a métrica do negócio. O conjunto de dimensões associadas à tabela de facto define o "grão" desta. Todas as medidas guardadas na tabela de facto devem deter o mesmo grão, de modo a que faça sentido a agregação/desagregação dos dados.

O modelo multi-dimensional pode ser implementado sobre bases de dados relacionais segundo diferentes esquemas, sendo os mais utilizados o *star schema* (Figura 1) ou o *snowflake schema* (Figura 2). O *star schema* é definido por uma tabela de facto e as diversas dimensões associadas a esta.

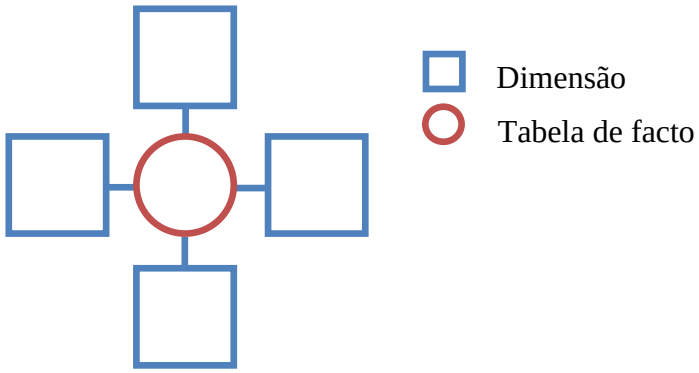


Figura 1 - *Star schema*.

O *snowflake schema* é, em parte, semelhante ao *star schema*, mas ao contrário do último as dimensões encontram-se normalizadas em diversas tabelas relacionadas.

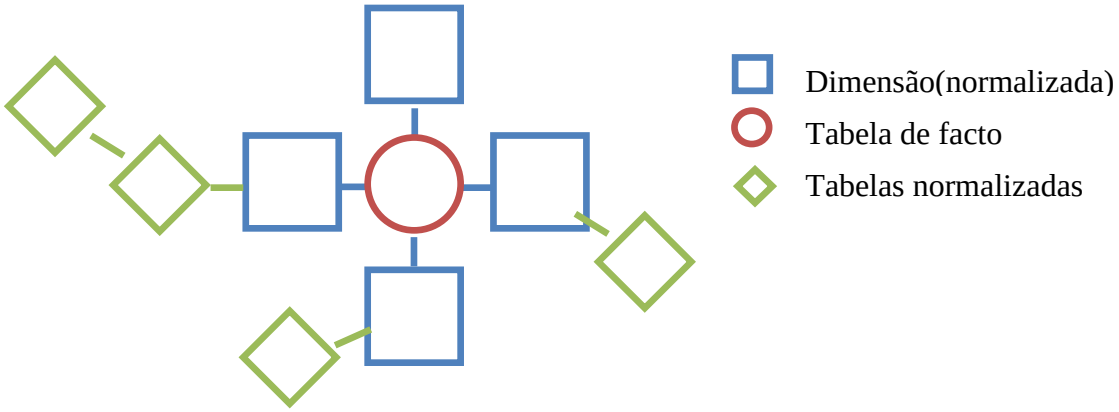


Figura 2 - *Snowflake schema*.

O conceito SOLAP vem adicionar dados geográficos ao modelo multi-dimensional, a partir dos quais são integrados mapas nas aplicações OLAP. Nos mapas podem ainda ser sobrepostos gráficos.

Os dados geográficos podem ser incluídos quer nas dimensões (originando o conceito **dimensões espaciais)**, quer nas tabelas de facto (surgindo o conceito **métricas espaciais**).

A introdução destes novos conceitos promove análises de possíveis correlações espaciais a partir da visualização de mapas. Além destes factores, estudos das ciências cognitivas mostram a superioridade dos mapas face a palavras e números [**3**].

Também a expressividade das interrogações deixa de estar limitada aos atributos alfanuméricos. Por exemplo, apresentar os resultados agregados por indústria para a seguinte análise: "Qual a quantidade de CO_2 emitida pelas indústrias a 3km de uma área protegida?".

Outra particularidade dos sistemas SOLAP, face aos sistemas OLAP, é o número de resultados que podem ser "consumidos" pelo utilizador. Nos sistemas OLAP as interrogações envolvem milhares de ocorrências [**6**], mas apresentam os dados muito sumarizados. Em contrapartida, nas ferramentas SOLAP os resultados podem não ser tão sumarizados e dependem do que é possível visualmente analisar.

Bédard apresentou em [**4**] as características fundamentais que um sistema SOLAP deve de incluir, sob três pontos de vista: (*i*) **visualização de dados** que engloba a sincronização entre as diferentes formas de visualização de dados (mapa, tabelas e gráficos), a construção de mapas temáticos adequados, a visualização de informação contextual e legenda interactiva; (*ii*) **exploração de dados** que inclui a navegação multi-dimensional em qualquer forma de visualização de dados, métricas calculadas e filtragem pelos atributos das dimensões; (*iii*) **estrutura dos dados** que abrange o suporte a diferentes formas geométricas e múltiplas representações para diferentes escalas.

Desde então, diferentes sistemas SOLAP têm sido desenvolvidos [**1**] [**7**] [**8**]. No geral, os sistemas desenvolvidos visavam incorporar algumas das características enunciadas acima. Outros sistemas tinham principalmente em vista a integração de métricas espaciais [**9**].

Tipicamente, o *software* que dá suporte aos sistemas OLAP é um *software* genérico que assenta numa abordagem multi-dimensional. Contudo, os diversos sistemas SOLAP têm sido desenvolvidos para um pré-determinado contexto, sem qualquer base de um *software* genérico. Existem, no entanto, duas linhas de trabalho de modo a desenvolver um sistema SOLAP genérico. A primeira está associada ao grupo de investigação de Bédard [**8**] e a segunda tem sido alcançada através de trabalhos de investigação e dissertações orientadas por Moura-Pires, J. [**10**] [**11**] [**1**].

A segunda linha de trabalho teve início com o trabalho de Rosa Matias (2006) [**10**], avançou com Vitorino e Caldeira (2008) [**11**] e terminou com o trabalho de Ruben Jorge (2009) [**1**]. Actualmente, o protótipo obtido consiste num sistema SOLAP genérico que engloba muitas das características enunciadas anteriormente: dá resposta à integração da informação espacial nas dimensões, de modo a que dados espaciais e dados não espaciais sejam utilizados em concordância; verifica a sincronização entre a visualização tabular e o mapa; permite a filtragem pelos atributos das dimensões; e suporta diversos casos de interacção que são a base em qualquer análise.

Embora o sistema genérico SOLAP obtido [**1**] signifique um bom avanço na área SOLAP, muitos são os desafios ainda em aberto: não prevê a integração de métricas espaciais, o processo de construção de mapas temáticos adequados baseia-se numa abordagem simples, os casos de

interacção propostos e implementados neste sistema genérico apenas anteviam atributos espaciais da mesma dimensão, não prevê situações onde o mapa poderá ficar desorganizado devido ao excesso de objectos geográficos (o que prejudica a visualização/análise de dados espaciais), entre outras.

Apesar de muitos desafios terem sido superados, muitos são aqueles ainda por se transpor. Assim, é dada continuidade ao trabalho de Ruben Jorge [**1**] com o objectivo de abordar alguns desses desafios.

Admitir casos de interacção com dois atributos espaciais de diferentes dimensões permitirá abrir um novo espectro de análises possíveis, que não são de todo realizáveis em casos de interacção com apenas uma dimensão espacial.

Em muitas áreas de actividade a necessidade (e utilidade) das análises recorrem simultaneamente a duas dimensões espaciais, como por exemplo, nos conjuntos de dados que englobem a noção de partidas e chegadas (ex: Companhias Aéreas, Ferroviárias, Transportes), telecomunicações, turismo, etc.

Para exemplificar os casos de interacção descritos anteriormente, considere um conjunto de dados para análise de voos, modelado segundo o *star schema* (Figura 3):

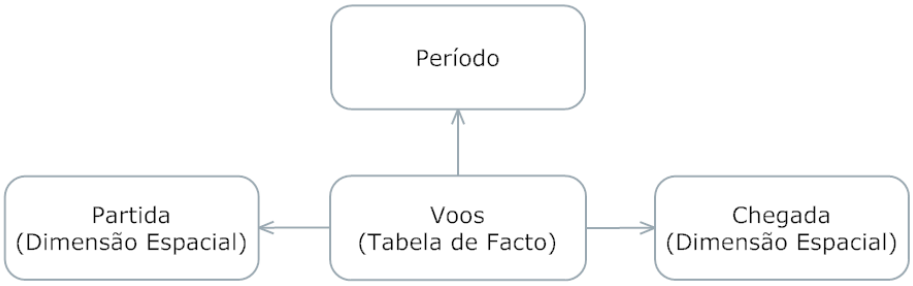


Figura 3 - Modelo multi-dimensional de um conjunto de dados de voos.

O modelo de dados contém as dimensões *Partida* e *Chegada*, as quais são dimensões espaciais que contêm um atributo com a localização geográfica de cada aeroporto.

Actualmente, nos sistemas [**1**] e [**8**] é possível analisar as métricas considerando apenas uma das dimensões espaciais. Imagine-se, para o exemplo anterior, que se quer analisar os voos entre os aeroportos. Uma opção seria considerar a dimensão espacial *Chegada*, e efectuar *slices* para cada valor da dimensão *Partida*, como se pode observar na Figura 4. A outra opção seria considerar o inverso.

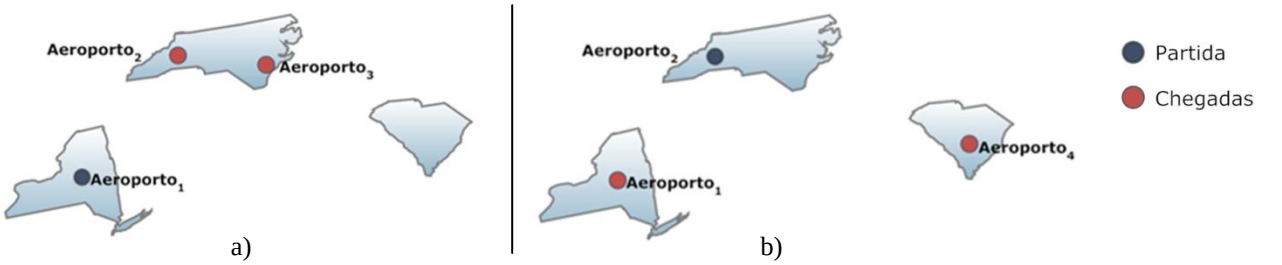


Figura 4 - Análise elementar de voos: a) partida = aeroporto$_1$; b) partida = aeroporto$_2$.

Esta solução criará múltiplos mapas (e forma tabular associada) para cada um dos valores distintos da dimensão *Partida*. Apesar de esta aproximação permitir ao utilizador analisar os dados para cada valor da dimensão *Partida*, esta não oferece uma forma compreensiva de observar as relações entre os dois atributos espaciais, tornando bastante difícil realizar análises comparativas entre eles.

Concluindo, nos sistemas SOLAP actuais não é possível obter uma visão global das relações entre dois atributos espaciais de diferentes dimensões para as métricas guardadas na tabela de facto.

Apesar da utilização dos mapas facultar novas e melhores análises, a verdade é que facilmente o mapa se poderá tornar desorganizado devido ao excesso de objectos espaciais. Em muitas situações poder-se-á verificar a desorganização que se observa na Figura 5.

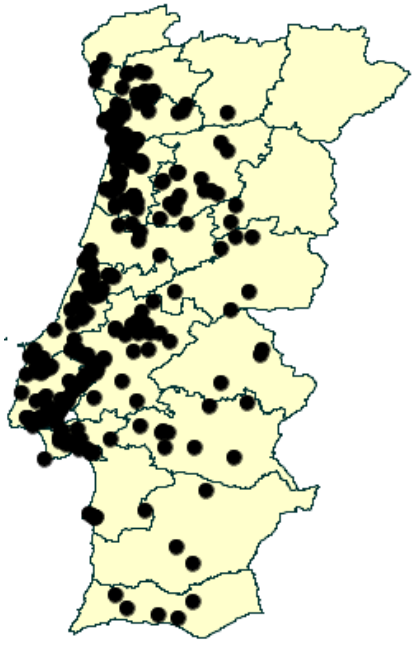


Figura 5 - Exemplo de um mapa desorganizado devido ao excesso de pontos.

Identificar as diferentes entidades no mapa pode ser um processo pouco evidente para o utilizador. Para resolver esta problemática é necessário reduzir o número de resultados e apresentá-los de forma adequada no mapa.

1.2 Objectivos e Contribuições

O objectivo desta dissertação é estender o modelo genérico SOLAP referido anteriormente [**1**], promovendo novas análises e mais intuitivas. Deste modo, temos como propósito oferecer soluções para alguns dos problemas ainda não resolvidos:

1. Suportar análises em que é necessária a utilização de dois atributos espaciais de diferentes dimensões;
2. Reduzir o número de resultados, quando se verifica excesso de objectos geográficos, através de algoritmos de agrupamento espacial.

Relativamente ao primeiro ponto, acreditamos que vem introduzir ao modelo genérico novas análises. Quanto ao segundo ponto, admitimos que este torne a "leitura" do mapa mais intuitiva.

Com o objectivo de dar solução a casos de interacção com dois atributos espaciais de diferentes dimensões, a principal contribuição deste trabalho é definir o seu modelo de interacção com base em [**1**].

Para resolver a questão do excesso de objectos espaciais no mapa, as contribuições são as seguintes:

- Introduzir duas formas de agrupamento espacial:
    - Agrupamento espacial *adhoc:*
        - Escolher o algoritmo mais adequado, dos analisados, para agrupamento espacial, quer para a forma geométrica *ponto*, quer para a forma geométrica *polígono*;
    - Definir um algoritmo de agrupamento espacial dinâmico com base nas hierarquias espaciais das dimensões espaciais.
- Integrar algoritmos de agrupamento espacial no modelo genérico SOLAP [**1**].

Apesar de termos em mente a introdução de algoritmos de agrupamento espacial nos diversos casos de interacção, teremos sempre em consideração a necessidade de introduzir o menor tempo de processamento adicional possível na interacção entre o utilizador e o sistema, de modo a que não viole o requisito já solucionado pelo modelo genérico SOLAP: o desempenho.

Além dos problemas anteriores, várias são as questões que se levantam na construção de mapas temáticos. Com o objectivo de garantir um papel efectivo do componente *mapa*, é inserido um gestor de estilos associado a um conjunto de regras baseadas na semiologia gráfica.

Como contributo final, pretendemos adicionar no protótipo desenvolvido em [**1**] a maioria das propostas realizadas nesta dissertação.

1.3 Estrutura da Dissertação

O restante documento está dividido em seis capítulos. No capítulo dois (Trabalho Relacionado) são apresentados os conceitos base inerentes à área SOLAP. Também são apresentadas algumas aplicações SOLAP e o estado de arte sobre agrupamento espacial. Ao longo do capítulo três (Extensão ao SOLAP+) são apresentadas as propostas para estender o modelo genérico SOLAP, que incluem dar suporte a casos com dois atributos espaciais de diferentes dimensões, integração de agrupamento espacial e a incorporação de um gestor de estilos. O capítulo quatro (Arquitectura) descreve a arquitectura do sistema, protocolo de comunicação e meta-modelo, realçando as alterações realizadas nesta dissertação. O capítulo cinco (Implementação) descreve da perspectiva de implementação os componentes que foram adicionados à arquitectura do sistema. A partir do capítulo seis (Caso de Estudo e Validação) são validadas as propostas realizadas apresentando exemplos de interacção utilizando o sistema SOLAP+. A dissertação termina com o capítulo sete (Conclusão e Trabalho Futuro), onde são apresentadas as conclusões e dadas direcções para trabalho futuro. O documento inclui um anexo que contém parte da definição do modelo de estilos.

Capítulo 2

Trabalho Relacionado

Este capítulo apresenta os conceitos base inerentes à área SOLAP, apresenta algumas aplicações SOLAP e o estado de arte do agrupamento espacial.

Ao longo deste capítulo será dada uma visão global dos trabalhos prévios realizados na área SOLAP. Inicialmente são apresentados conceitos base necessários à compreensão das secções seguintes. Estes conceitos estão relacionados com os fundamentos dos sistemas de suporte à decisão OLAP. Na secção seguinte, são apresentados alguns dos sistemas SOLAP desenvolvidos, de modo a obter uma visão das suas características e possíveis limitações. No final, são analisados os diferentes algoritmos de agrupamento espacial de modo a atingir um dos objectivos desta dissertação: agrupar pontos e polígonos.

2.1 Conceitos Base

Ao longo desta secção são abordadas as seguintes questões ou conceitos:

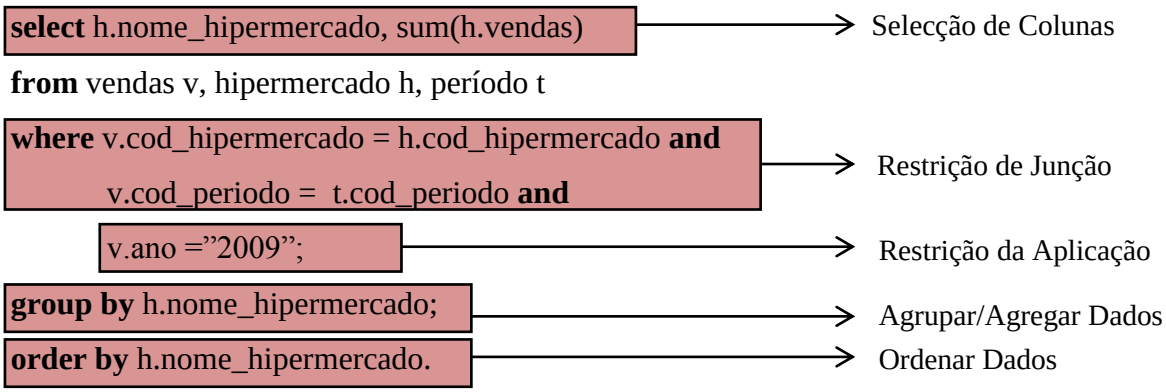
- De que forma é processada uma interrogação OLAP;
- Como são apresentados, ao utilizador, os resultados originados por uma interrogação;
- Tipos de dimensões espaciais;
- Hierarquias.

2.1.1 Interrogação OLAP

Nesta secção, são consideradas apenas implementações dos modelos multi-dimensionais sobre bases de dados relacionais.

Figura 6 - Modelo simplicado do conjunto de dados de vendas.

Considere a seguinte interrogação OLAP, sobre um conjunto de dados de vendas para grandes superfícies comerciais modelado segundo o *star schema* (Figura 6).



Dada uma interrogação Q, esta é processada da seguinte forma: primeiro, as restrições da aplicação são processadas para cada dimensão, em que cada dimensão produz um conjunto de chaves candidatas. Posteriormente, todas as chaves candidatas são concatenadas (produto cartesiano) de modo a obter o conjunto de chaves que podem ser pesquisadas na tabela de facto. Todas as chaves que corresponderem na tabela de facto são agrupadas e agregadas.

O resultado de uma interrogação OLAP consiste num conjunto de linhas que contém os atributos das dimensões seleccionados e as respectivas métricas agregadas. Um possível conjunto de linhas para a interrogação anterior é o seguinte:

Nome do Hipermercado	Total de Vendas (Euro)
X	4.000.000.000
Y	5.000.000.000
Z	1.000.000.000

Tabela 1 - Conjunto de linhas resultante da *query* OLAP anterior.

Tipicamente, de uma interrogação OLAP resulta um conjunto de linhas de pequena dimensão, apesar de estarem envolvidas, em geral, milhares de linhas de dados. O facto de o conjunto de linhas ser reduzido é fundamental para os utilizadores dos sistemas OLAP.

Para efeitos de análise, o conjunto de linhas não deve ser disponibilizado ao utilizador na forma original com que é retornado da interrogação. A apresentação dos dados é realizada, por norma, com recurso a tabelas pivô, com o objectivo de tornar as análises dos utilizadores mais evidentes.

Suponha que se quer analisar as vendas por semestre. Com este objectivo era adicionado o atributo *semestre* à análise anterior. A esta operação designa-se de uma operação *drill-down* pois os dados são desagregados e observados a um nível mais detalhado. Em resultado desta operação, os dados são disponibilizados em forma pivô (Figura 7):

	Semestre	
	Primeiro	Segundo
Nome do Hipermercado	Total de Vendas	Total de Vendas
X	1.000.000.000	3.000.000.000
Y	3.000.000.000	2.000.000.000
Z	500.000.000	500.000.000

Figura 7 - Tabela pivô em resultado da operação de desagregação.

2.1.2 Hierarquias

Uma dimensão é composta por atributos que permitem explorar as métricas, a partir de diferentes pontos de vista. Estes atributos podem formar uma *hierarquia*. Existem diferentes tipos de hierarquias. As mais comuns são aquelas que têm uma representação em árvore, isto é, dados dois níveis consecutivos h_1 e h_2, para cada valor de h_1 existe apenas um valor de h_2 (assumindo que h_2 está a um nível superior de granularidade em relação a h_1). Exemplos:

1. mês, trimestre, semestre, ano;
2. loja, freguesia, concelho, distrito.

Com a introdução de informação espacial nas dimensões surgiram novos tipos de hierarquias: as *hierarquias espaciais*. Um possível exemplo destas hierarquias seria o exemplo 2 associado a dados espaciais, em vez de estar associado a dados semânticos.

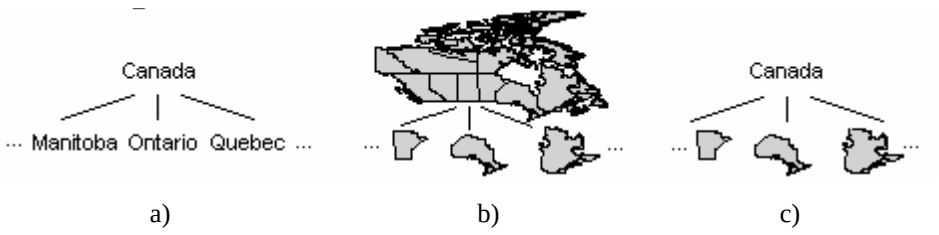


Figura 8 - Tipos de dimensões : a) não geométrica; b) geométrica; c) mista.

As dimensões podem ser classificadas com base no tipo de dados presente em cada nível: (*i*) **não geométrica** (quando todos os níveis contêm dados alfanuméricos); (*ii*) **geométrica** (quando todos os níveis contêm dados espaciais); (*iii*) **mista** (quando existem níveis com dados alfanuméricos e níveis com dados espaciais). Os diferentes tipos de dimensões estão ilustrados na Figura 8 (obtida de [**3**]).

2.2 Sistemas SOLAP

Desde que surgiu o conceito SOLAP, alguns sistemas têm sido desenvolvidos. Apesar de uma década de investigação, apenas nos últimos anos se tem verificado o aparecimento de aplicações comerciais, como é o caso da aplicação JMap [**8**]. Outras aplicações têm sido desenvolvidas num âmbito académico.

Nas secções subsequentes serão apresentadas três aplicações SOLAP. Cada uma delas com diferentes características, permitindo apresentar uma visão global do estado actual das aplicações SOLAP.

2.2.1 SOVAT

A aplicação SOVAT [7] [**12**] foi desenvolvida para auxiliar a comunidade da área da saúde pública nas suas análises.

A *interface* (Figura 9) está dividida em quatro áreas distintas: (*i*) mapa; (*ii*) zona do gráfico; (*iii*) painel para construir a interrogação; (*iv*) painel para executar interrogações especiais.

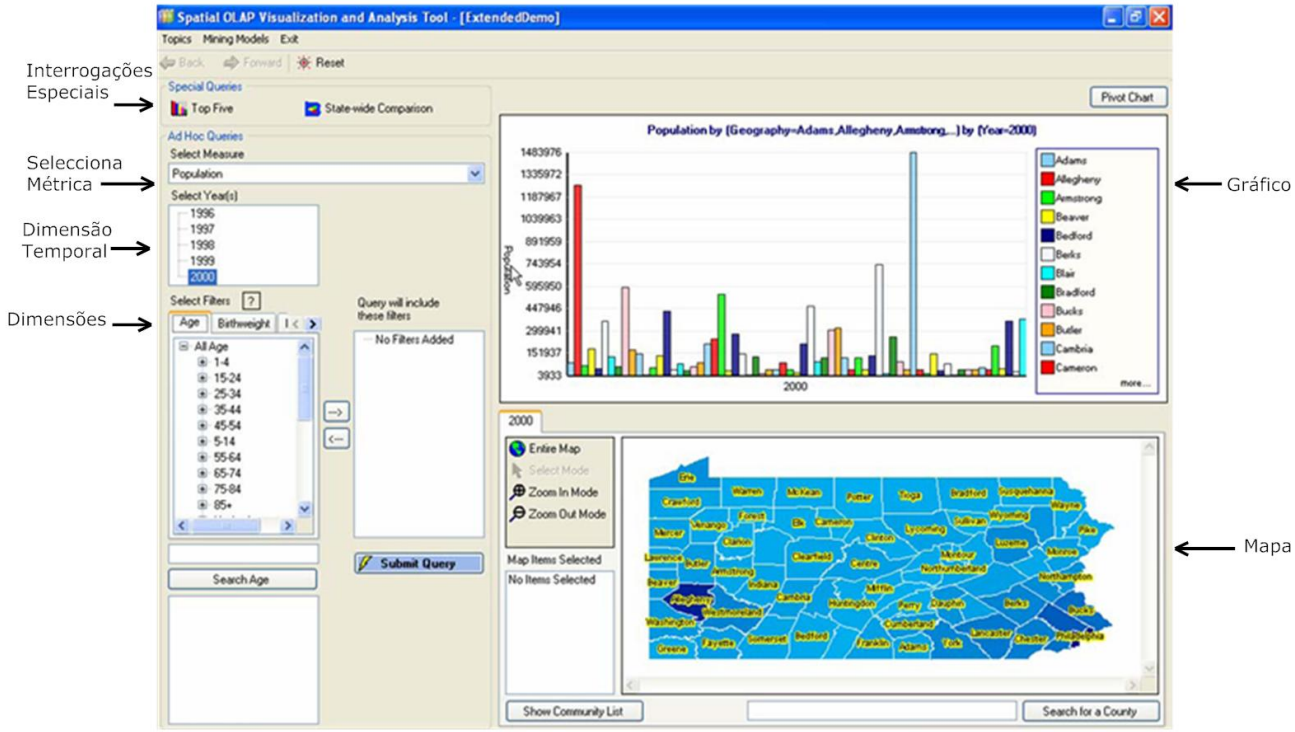


Figura 9 - *Interface* da aplicação SOVAT.

No painel de construção da interrogação o utilizador escolhe a métrica que quer analisar. A dimensão temporal é colocada abaixo da *combo box* onde é seleccionada a métrica (neste caso a dimensão é *ano*). Cada *tab* corresponde a uma dimensão, onde o utilizador pode realizar filtragens dos dados. Adicionalmente suporta as operações de *drill-up* e *drill-down* sobre os atributos espaciais. Esta aplicação não suporta carregamento de outros conjuntos de dados.

2.2.2 JMap

A aplicação JMap [**8**] foi inicialmente desenvolvida pelo grupo de investigação de Bédard, mas nos últimos anos tem sido desenvolvida pela empresa *KHEOPS Technologies*.

A *interface* da aplicação (Figura 10 e Figura 11) está dividida em duas áreas: (*i*) uma área fixa, onde se encontram as ferramentas (ex: aumentar zoom, diminuir zoom, etc.), a descrição do modelo

multi-dimensional, onde o utilizador pode escolher os atributos que quer analisar; (*ii*) uma área que permite o utilizador escolher qual a forma de visualização dos dados (tabela, mapa ou gráfico).

Engloba muitas das características que Bédard considera importantes num sistema SOLAP (referidas na secção 1.1), nomeadamente a sincronização entre as diferentes formas de visualização de dados, inclui operações de *drill-up* e *drill-down*, suporta diferentes formas geométricas, etc.

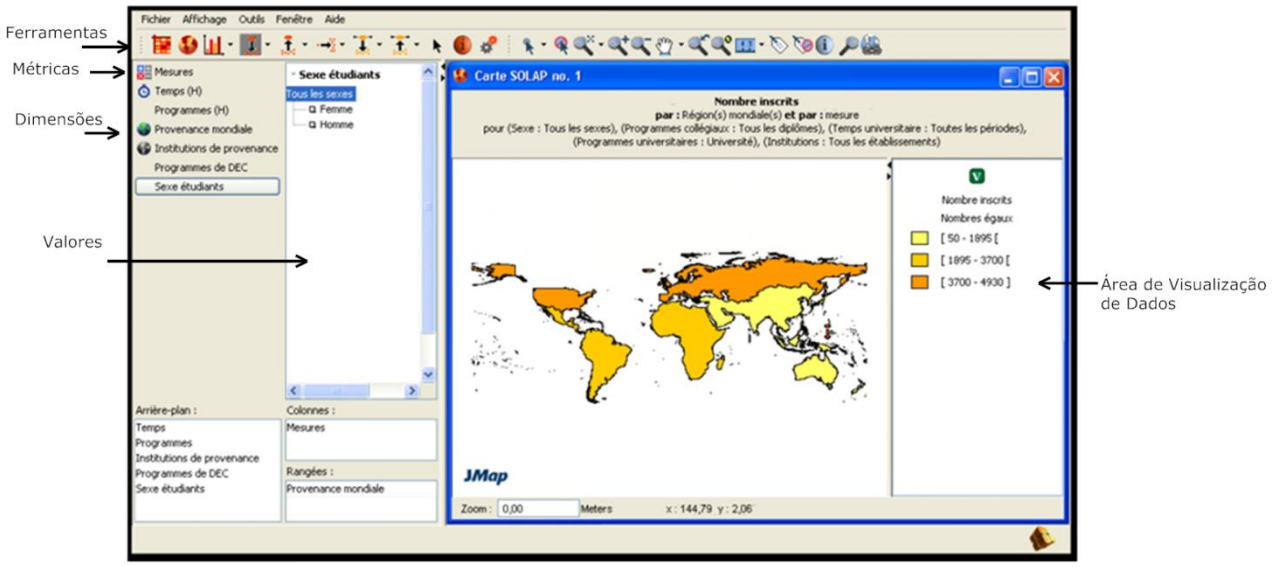


Figura 10 - *Interface* da aplicação *JMap* ao visualizar o mapa.

A tabela utilizada para apresentar os dados é uma tabela pivô, onde as colunas e as linhas correspondem aos atributos que estão seleccionados nos painéis *Colonnes* e *Rangées* (Figura 11).

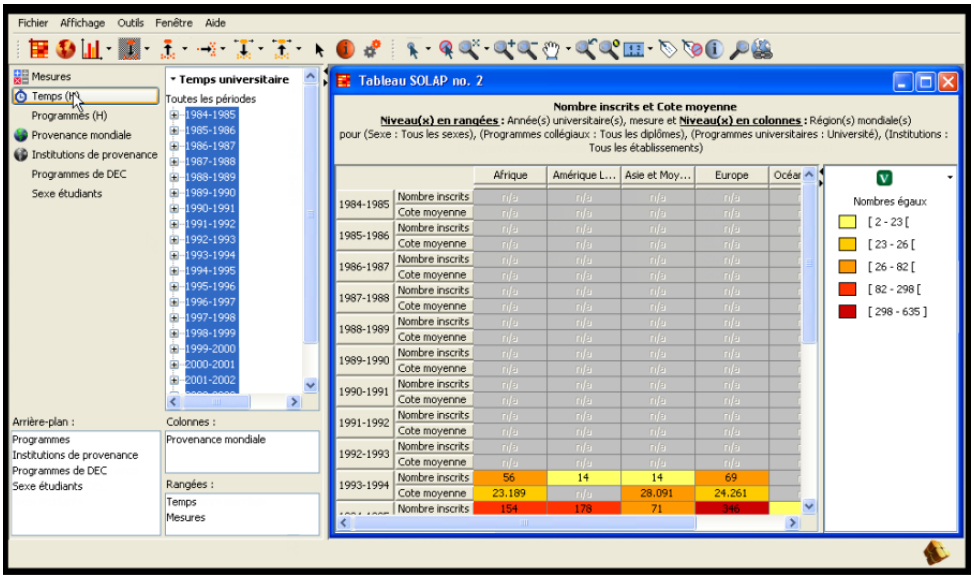


Figura 11 - *Interface* da aplicação *JMap* ao visualizar a tabela.

Este sistema não foi concebido para um pré-determinado contexto. Tem uma ferramenta desenvolvida que permite o administrador do sistema construir o modelo desejado (escolher métricas, dimensões, etc.).

2.2.3 SOLAP+

O SOLAP+ resultou da linha de trabalho sob a orientação de Moura-Pires, J., realizado no departamento de informática na UNL – FCT, que culminou em [**1**].

A *interface* (Figura 12) do sistema está dividida em seis áreas: (*i*) componente mapa; (*ii*) tabela de suporte; (*iii*) tabela de detalhe; (*iv*) descrição do modelo multi-dimensional (painel da direita); (*v*) área de *slices*; (*vi*) área para configuração de aspectos no mapa (*map control*).

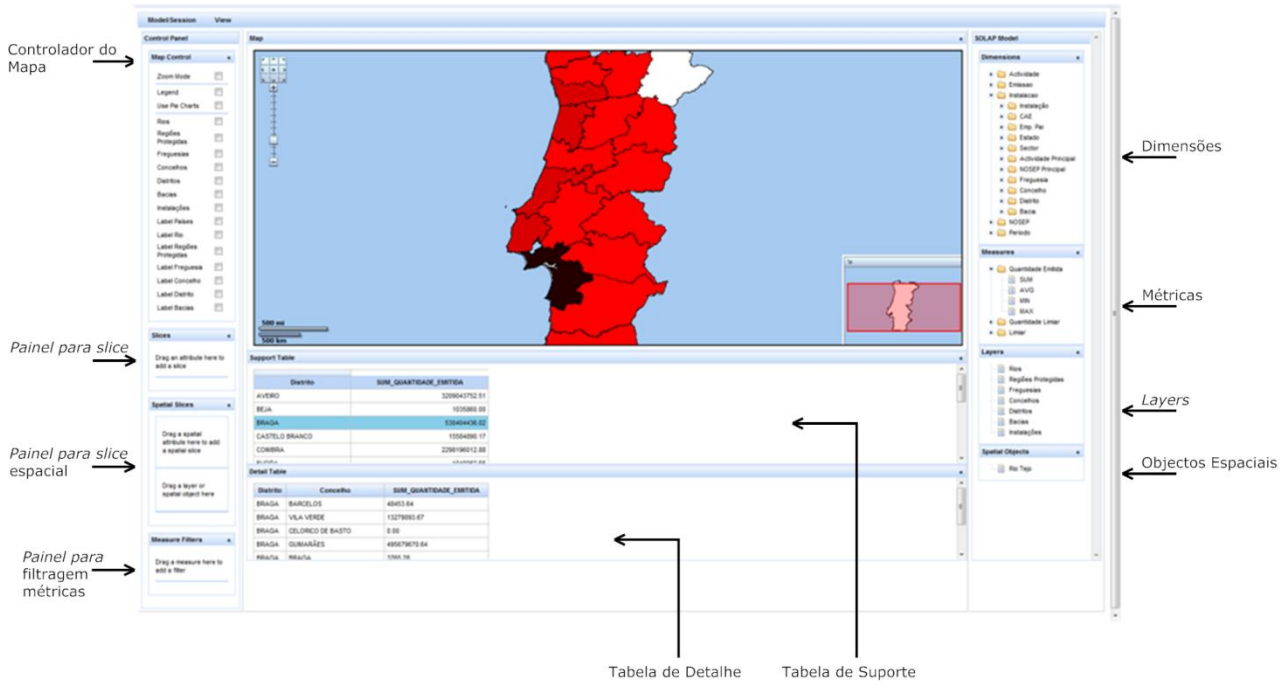


Figura 12 - *Interface* do modelo de genérico SOLAP.

A tabela de detalhe disponibiliza os dados a um nível de granularidade inferior, comparativamente com os dados que se encontram na tabela de suporte. O sistema possibilita operações de filtragem por dados não espaciais, espaciais (*spatial slices*) ou por métricas.

Uma propriedade fundamental neste sistema é a relação de 1:1 entre a representação no mapa e a tabela de suporte. Esta propriedade é mantida em qualquer situação. Para demonstrar a importância desta propriedade, vamos verificar as consequências da ausência dela:

- Múltiplas linhas na tabela de suporte associadas a um objecto gráfico;
- Múltiplos objectos gráficos associados a uma linha na tabela de suporte.

No primeiro caso, apresentar diversas linhas da tabela de suporte associadas a um único objecto gráfico leva à seguinte situação (Figura 13):

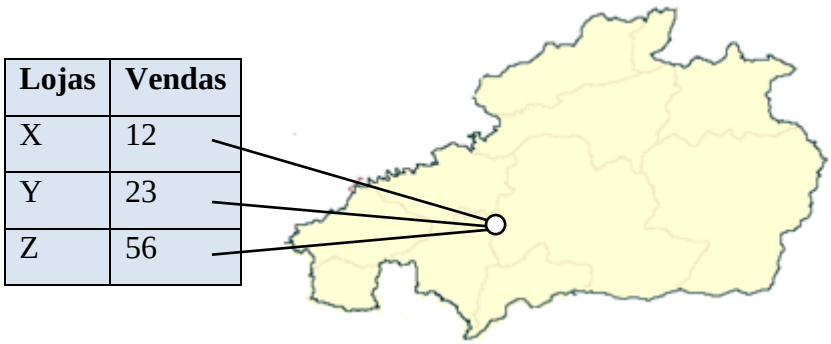


Figura 13 - Múltiplas linhas associadas a um objecto gráfico.

Este contexto pode levar a mapas temáticos confusos, pois seria necessário apresentar os dados das diferentes linhas associadas ao objecto geográfico no mapa. Além disso, o processo de associação entre os dados presentes na tabela de suporte e o objecto geográfico pode ser complicado, caso as linhas pertencentes ao mesmo objecto estejam dispersas pela tabela [**1**].

O segundo caso, a apresentação de diversos objectos geográficos associados a uma linha na tabela de suporte, é ilustrado na Figura 14.

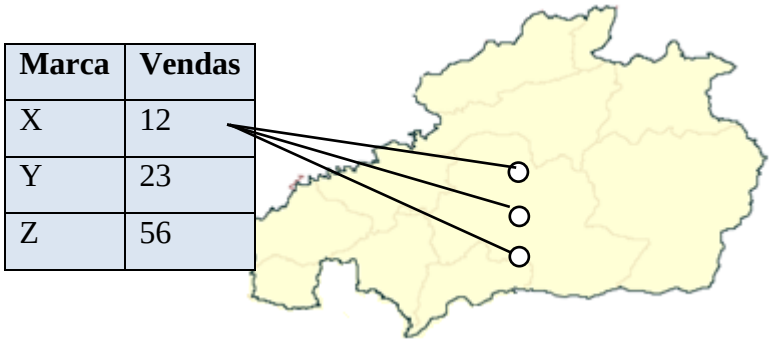


Figura 14 - Múltiplos objectos geográficos associados a uma linha da tabela de suporte.

Nesta situação, o utilizador não será capaz de observar de que forma os diferentes objectos geográficos contribuem para a informação guardada na linha da tabela de suporte [**1**]. Por outro lado, se considerarmos objectos geográficos distantes uns dos outros, o processo de associação entre a tabela de suporte e o mapa tornar-se-á bastante complicado.

2.2.4 Discussão dos Sistemas SOLAP

A aplicação SOVAT representa as típicas aplicações SOLAP que são construídas para um determinado contexto. Ainda assim, verifica a integração de dados espaciais (nas dimensões) e a sincronização entre o mapa e o gráfico.

Tanto o JMap como o SOLAP+ suportam carregamento de diferentes conjuntos de dados e, de uma maneira geral, incorporam as características que Bédard reconhecia como imprescindíveis numa aplicação SOLAP. No entanto, têm conceitos de interacção diferentes. Relativamente ao SOLAP+, este permite a visualização em simultâneo do mapa e da tabela pivô, mantendo sempre a relação de 1:1 entre estes componentes. O mesmo conceito não se verifica na aplicação JMap. Por outro lado, o último também não suporta a visualização de dois níveis de granularidade em simultâneo, o que pode ser útil em alguns casos. Já o SOLAP+ verifica dois níveis de granularidade para a apresentação de dados: tabela de suporte e tabela de detalhe.

Ainda assim, apesar de ambas as aplicações terem realizado um avanço significativo, no geral as aplicações SOLAP verificam as seguintes limitações: (*i*) não suportam métricas espaciais; (*ii*) não suportam análises na presença de dois atributos espaciais de diferentes dimensões; (*iii*) não têm quaisquer mecanismos de agrupamento espacial ou não espacial; (*iv*) não suportam uma legenda interactiva; (*v*) outras questões em aberto. Todas estas questões são também referidas pelos autores de [**3**] [**1**] [**13**].

2.3 Agrupamento Espacial

O processo de agrupamento (*clustering*) consiste na aprendizagem automática não supervisionada, com o objectivo de organizar os dados em grupos de tal forma que: (*i*) exista uma forte similaridade entre os elementos pertencentes ao mesmo grupo; (*ii*) exista uma fraca similaridade dos elementos pertencentes a grupos diferentes [**14**]. Ao contrário da aprendizagem supervisionada, a descoberta da estrutura dos dados é produzida directamente a partir destes, com o intuito de identificar os grupos e o número de grupos. Assim, um grupo é uma colecção de dados que são semelhantes, relativamente uns aos outros, e são dissimilares aos objectos que se encontram em outros grupos.

Este processo tem sido utilizado em inúmeras aplicações, tais como na biologia, na medicina, no reconhecimento de padrões, em análise de dados, entre outras. Na biologia pode ser utilizada para definir uma taxonomia de plantas e animais; nos negócios (análise de dados) pode ajudar os comerciantes a categorizar diferentes tipos de clientes, com base nos seus padrões de compra, para que possam tomar diferentes estratégias consoante o tipo de clientes. Em termos gerais, esta técnica

pode ser utilizada para obter uma visão da distribuição dos dados, observar as características de cada grupo e focar num determinado grupo para uma análise posterior.

O facto da técnica de análise de agrupamento criar uma abstracção representativa dos dados originais pode trazer diversas utilidades, nomeadamente a sumarização. Quando se está a lidar com um conjunto de dados de elevada dimensão pode ser vantajoso utilizar esta técnica para reduzir o número de elementos observáveis e representativos dos dados originais.

Ao longo dos anos, com a crescente utilização de informação geográfica nas bases de dados, surgiu uma nova forma de agrupamento, designada de agrupamento espacial (*spatial clustering*). O processo de agrupamento espacial é uma técnica em tudo semelhante ao agrupamento. No entanto, esta técnica pode ser utilizada com base em combinações de atributos não espaciais, espaciais, proximidade de objectos ou eventos no espaço, no tempo ou ambos (espácio-temporal). As muitas aplicações desta técnica resultaram num tremendo crescimento da área de agrupamento espacial. Inerente a este crescimento surgiu um elevado número de algoritmos propostos na literatura.

Na área de agrupamento espacial é necessário ter em conta as formas geométricas. Em geral, a mais abordada é o ponto. Tipicamente, a localização de uma entidade ou evento é representado por um ponto. No entanto, também é necessário ter em conta outras formas geométricas, nomeadamente linhas e polígonos. As linhas são normalmente utilizadas para representar rios/estradas e os polígonos utilizados para representar regiões.

Dada a enorme variedade desta área de agrupamento, nesta dissertação apenas nos vamos focar no agrupamento espacial com base na proximidade de objectos no espaço. O agrupamento da forma geométrica linha não entra no escopo desta dissertação. Recordo que o âmbito desta dissertação é reduzir o número de resultados quando se verifica o excesso de objectos geográficos no mapa, e, em geral, as formas geométricas mais frequentes nos conjuntos de dados são o ponto e o polígono.

2.3.1 Requisitos

Muitos são os desafios que se colocam quando se quer desenvolver uma técnica de agrupamento espacial. Actualmente, existem já trabalhos que identificaram os diversos requisitos [**15**] [**14**]. São eles:

Escalabilidade

O algoritmo de agrupamento deve ser capaz de lidar com bases de dados que contenham elevadas quantidades de informação. Deste modo, os algoritmos devem ter complexidades temporais e

espaciais baixas. Por outro lado, num contexto SOLAP, é importante introduzir o mínimo de tempo de processamento, com o objectivo de manter as análises fluidas.

Versatilidade

O algoritmo deve ser capaz de lidar com diferentes tipos de objectos. Por exemplo, possibilitar tanto o agrupamento de pontos como de polígonos. Por outro lado, o tamanho do conjunto de dados não deve influenciar a qualidade dos resultados obtidos por um algoritmo.

Capacidade de identificar grupos com formas irregulares

A capacidade de identificar grupos com formas arbitrárias é bastante importante. Muitos dos algoritmos apenas conseguem identificar grupos com forma esférica. No entanto, é frequente existirem grupos com formas geométricas irregulares.

Mínimo conhecimento do domínio para determinar os parâmetros de entrada

Esta característica é das mais difíceis de ser alcançada pelos algoritmos. Em muitos casos existem mesmo algoritmos que requerem ter conhecimento *a priori* do número de grupos a ser determinado. Outros algoritmos têm parâmetros chave, pouco intuitivos, que fazem destes algoritmos impraticáveis em aplicações no mundo real.

Robusto na presença de ruído

O resultado de um algoritmo de agrupamento deve ser independente da quantidade de *outliers* que possam existir no conjunto de dados.

Insensível à ordem de entrada dos dados

O resultado de um algoritmo de agrupamento deve ser igual, independentemente da ordem de entrada dos dados.

2.3.2 Algoritmos: Agrupamento de Pontos

Existe uma enorme variedade de algoritmos de agrupamento de pontos. Como é ilustrado na figura abaixo (Figura 15), os algoritmos podem ser classificados em quatro categorias: (*i*) método por partição; (*ii*) método hierárquico; (*iii*) método baseado na densidade; (*iv*) método baseado em grelha.

Na Figura 15 não estão presentes todos os algoritmos existentes. Nesta dissertação foram apenas analisados alguns algoritmos, e esses correspondem àqueles que estão destacados na Figura 15. Em resultados de estudos prévios [**16**] [**15**] [**14**] [**17**], para cada tipo de método, são analisados os

algoritmos que se apresentam como os potenciais melhores algoritmos para atingir o objectivo desta dissertação, à excepção de k-Means e k-Medoid que são introduzidos por motivos históricos. A forma geométrica considerada por estes algoritmos é o ponto.

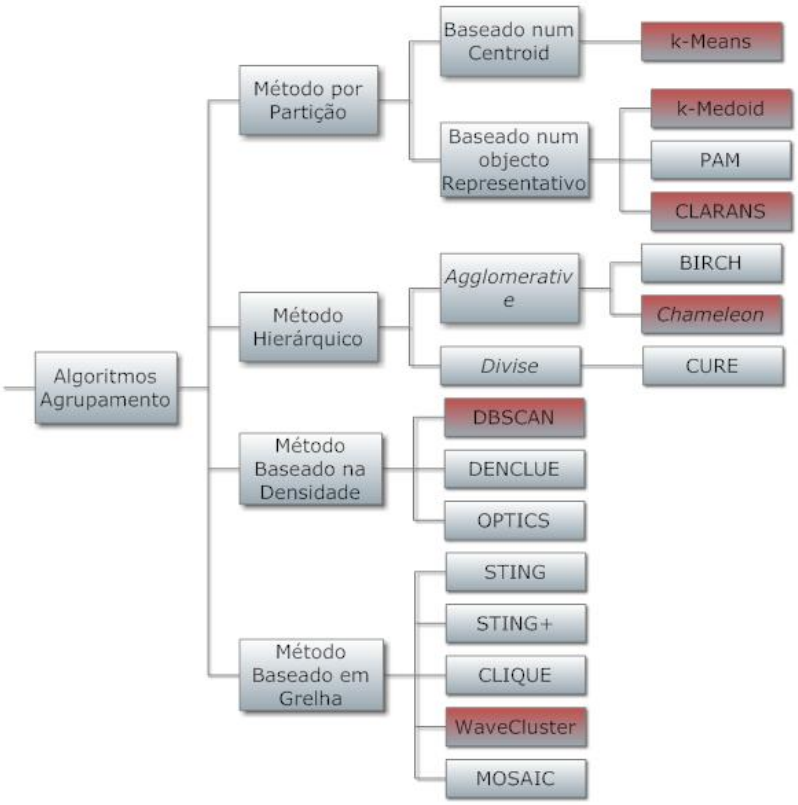


Figura 15 - Taxonomia dos algoritmos de agrupamento.

A escolha dos algoritmos analisados foi também efectuada de modo a ter uma visão mais aprofundada de cada um dos métodos existentes.

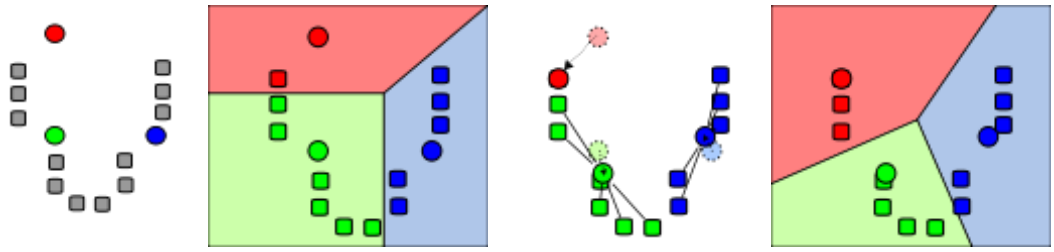
2.3.2.1 Métodos Por Partição

Em geral, os métodos por partição dividem o conjunto de dados original em *k* partições, onde cada partição corresponde a um grupo. O processo de particionamento tem como objectivo obter o óptimo de um determinado critério. O critério de agrupamento tipicamente utilizado é a *soma dos erros quadráticos*[1] [**17**]. Após a classificação dos *k* grupos, têm de ser verificadas as seguintes condições: (*i*) cada grupo deve conter pelo menos um objecto; (*ii*) cada objecto pertence a um ou nenhum grupo.

[1] Informalmente, este critério mede o quão bem um determinado conjunto de dados é representado pelo respectivo centróide do grupo.

k-Means

O algoritmo *k-Mean* [**18**] é uma técnica baseada em pontos representativos (centróides) que consistem em centros de massa de cada grupo. O algoritmo *k-Means* tem como parâmetro de entrada o valor *k*, que representa o número de partições em que o algoritmo irá particionar o conjunto de dados inicial. O algoritmo procede da seguinte forma: primeiro escolhe aleatoriamente *k* objectos do conjunto de dados como pontos iniciais e respectivos centros de massa de cada partição; de seguida, constrói os *k* grupos ao atribuir a cada objecto o ponto representativo mais similar; depois actualiza os pontos representativos de cada grupo, como pode ser visualizado na figura seguinte (Figura 16). Este processo é repetido enquanto a qualidade do particionamento é melhorada (utilizando um determinado critério), isto é, até que os pontos representativos se mantenham inalterados.



i) *ii*) *iii*) *iv*)

Figura 16 - Ilustração do Algoritmo *k-Means* com *k=3*.

Considere o exemplo da Figura 16. Na fase inicial, é feita a escolha dos *k* objectos iniciais (no exemplo $k = 3$) representativos de cada partição; a fase seguinte consiste no agrupamento resultante, associando cada ponto ao centro de massa mais próximo; depois são actualizados os pontos representativos, visto que a qualidade do agrupamento melhora; por fim, quando é obtida a convergência, o algoritmo termina.

A complexidade temporal do algoritmo em cada iteração é linear. A cada iteração, com *k* o número de grupos e *n* o tamanho do conjunto de objectos, a complexidade temporal define-se da seguinte forma: $O(k * n)$. Apesar de ser um algoritmo relativamente escalável, contém diversos aspectos negativos: é necessário especificar o número de grupos; não permite identificar grupos com formas arbitrárias; é sensível a *outliers*; e a escolha inicial dos pontos representativos influencia o resultado final do algoritmo.

k-Medoid

O algoritmo *k-Medoid* [**16**] é semelhante ao algoritmo *k-Means*. No entanto, em vez dos pontos representativos serem centros de massa, são pontos pertencentes ao conjunto de dados que mais se aproximam do centro de massa de cada partição (*medoid).*

Ao contrário do algoritmo anterior, a complexidade deste algoritmo é quadrática em cada iteração. Seja *n* a dimensão do conjunto de objectos e *k* o número de partições resultantes. A complexidade temporal, em cada iteração, corresponde a $O(k * (n-k)^2)$. Este facto deve-se às trocas dos pontos representativos que são necessárias efectuar. Deste modo, verifica-se que é um algoritmo pouco escalável, mantém a necessidade de especificar o número de grupos e é incapaz de computar partições arbitrárias. O facto dos pontos representativos não serem centros de massa faz com que este algoritmo se torne mais robusto na presença de *outliers*, comparativamente com o algoritmo anterior (*k-Means*).

CLARANS

O algoritmo CLARANS [**19**] surgiu da necessidade de lidar com grandes conjuntos de objectos. Deste modo, evoluiu-se do algoritmo PAM (*k-Medoid*) para o algoritmo CLARA, e mais tarde obteve-se CLARANS.

A ideia chave do algoritmo CLARA é a seguinte: em vez de encontrar os objectos representativos de cada grupo para todo o conjunto de dados, CLARA extrai uma amostra do conjunto de dados para encontrar os *medoids* desta. Depois, a cada objecto do conjunto de dados, atribui o *medoid* mais similar. Embora o algoritmo CLARA tenha obtido melhores tempos de resposta que PAM, pode não obter os melhores resultados. O facto do algoritmo CLARA se basear em amostras pode prejudicar o resultado, devido aos *medoids* resultantes da amostra.

Deste modo, para melhorar os tempos de resposta e a qualidade dos resultados conseguidos pelo algoritmo CLARA, foi proposto o algoritmo CLARANS. Ao contrário de CLARA, CLARANS não se limita a qualquer pré-determinada amostra. Conceptualmente, o processo pode ser visto como uma pesquisa aleatória num grafo em que cada nó corresponde a uma possível solução (um conjunto de *k* representativos). O algoritmo tem como parâmetros de entrada *maxneighbor* e *numlocal*. O primeiro corresponde ao número máximo de vizinhos de um nó que podem ser analisados, e o segundo representa o número máximo de mínimos locais que podem ser obtidos. O processo inicia através da escolha aleatória de um nó *N* do grafo. Após a escolha do nó é verificada uma amostra dos vizinhos de *N* cuja dimensão tem de ser menor ou igual que $maxneighbor$. Se for identificado um melhor vizinho, CLARANS move-se para esse nó e repete o processo. Caso contrário, considera o nó local como um mínimo local. Após serem encontrados *numlocal* mínimos locais, o algoritmo termina e retorna o melhor mínimo local.

Apesar de este algoritmo melhorar os tempos de resposta, a complexidade temporal mantêm-se quadrática $O((k * n)^2)$, onde k é o número de grupos e n corresponde à dimensão do conjunto de dados. Assim, é possível reter que as melhorias obtidas não são ainda satisfatórias na presença de grandes conjuntos de dados. No entanto, a qualidade é melhorada uma vez que o algoritmo escolhe uma amostra em cada etapa de pesquisa por vizinhos. É também importante referir que deixou de ser necessário especificar o número de grupos.

2.3.2.2 Métodos Hierárquicos

O método hierárquico cria uma decomposição hierárquica dos conjuntos de objectos. Este método pode ainda ser subdividido em duas categorias: aglomerativas ou divisivas, consoante a decomposição hierárquica seja efectuada de baixo para cima ou vice-versa, respectivamente, como é ilustrado na Figura 17, retirada de [**16**].

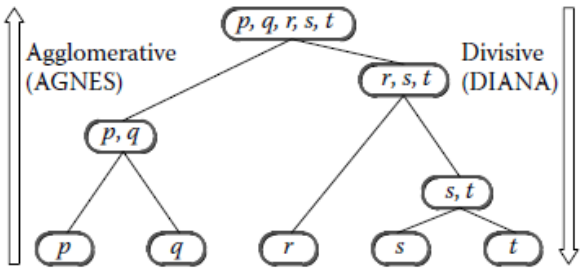


Figura 17 - Agrupamento *Agglomerative* e *Divisive*.

De um algoritmo hierárquico, em geral, resulta um dendograma que consiste numa árvore binária que permite visualizar a estrutura hierárquica dos agrupamentos. Neste tipo de algoritmos não é necessário especificar o número k de grupos, todavia, é necessário definir a condição de paragem para que o processo de *junção* ou *divisão* seja terminado. A decisão de agregar ou desagregar pontos é sempre uma decisão chave neste processo, pois todo o restante processo ir-se-á basear nas operações efectuadas anteriormente. Com o intuito de corrigir estas situações foi introduzia a técnica designada de multi-fase, que consiste em incorporar, ao método hierárquico, outras técnicas de agrupamento. O algoritmo analisado baseia-se na técnica multi-fase.

Chameleon

O algoritmo Chameleon [**20**] foi proposto para resolver algumas das limitações dos algoritmos aglomerativos. Em [**20**], o autor afirma que alguns algoritmos já propostos ignoravam a inter-conectividade de objectos em dois grupos, enquanto que outros ignoravam a proximidade de dois grupos.

Inicialmente os dados são representados em forma de grafo. O grafo utilizado é o *k-nearest-neighbor*[2], onde cada vértice representa um objecto. Deste modo, o algoritmo *Chameleon* é executado em duas fases, como pode ser observado na Figura 18, baseada em [**20**].

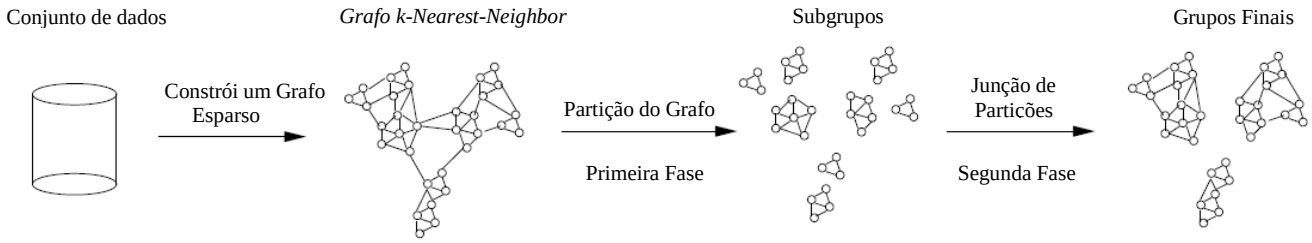


Figura 18 - Processo multi-fase do algoritmo *Chameleon.*

Durante a primeira fase, *Chameleon* utiliza um algoritmo de partição de grafos para agrupar os dados num número relativamente grande de subgrupos, de tal forma que a soma dos pesos dos arcos das diferentes partições seja minimizada (o peso dos arcos corresponde ao valor de similaridade entre os objectos). O processo de bipartição termina quando o maior sub*g*rupo contiver menos vértices do que o especificado pelo parâmetro $minsize$. Na segunda fase, determina-se a similaridade entre cada par de subgrupos C_i e C_j de acordo com a sua relativa inter-conectividade $R_I(C_i, C_j)$ e a sua relativa proximidade $R_c(C_i, C_j)$. Deste modo, utiliza um algoritmo aglomerativo hierárquico para encontrar os grupos através da união dos subgrupos, em que se verificam valores elevados para a relativa inter-conectividade (R_I) e para a relativa proximidade (R_c) entre eles.

Este algoritmo permite duas formas de ser parametrizado. Uma delas é a definição do T_{ri} (limiar de relativa inter-conectividade) e do T_{rc}(limiar de relativa proximidade), em que dois subgrupos serão unidos se se verificar $R_I(C_i, C_j) \geq T_{ri}$ e $R_c(C_i, C_j) \geq T_{rc}$. A outra forma de parametrização é o utilizador especificar o parâmetro α. Se $\alpha > 1$, o algoritmo dará mais importância à relativa proximidade. Caso contrário, dará mais importância à relativa inter-conectividade.

A complexidade do algoritmo é quadrática. Se n for a dimensão do conjunto de dados, a complexidade temporal é $O(n^2)$. Este algoritmo escala até um número relativamente grande de objectos. Contudo, assume que o conjunto de objectos está todo guardado em memória o que limita a sua escalabilidade. É um algoritmo que tem potencial para descobrir grupos com formas arbitrárias. Mas, no melhor caso, o algoritmo depende de diversos parâmetros: k (número de vizinhos a

[2] Seja G = (V, E), onde cada vértice v_i no grafo representa um objecto do conjunto de dados e cada arco entre v_ie v_j representa a similaridade entre os objectos. Num grafo *k-nearest-neighbor* apenas existe um arco de v_i para v_j se v_j estiver entre os *k* vizinhos mais próximos de v_i.

considerar no *k-nearest-neighbor*), $minsize$ (condição de paragem para o particionamento do grafo) e α. Este facto requer que haja algum conhecimento do domínio dos dados para determinar os valores adequados dos parâmetros.

2.3.2.3 Métodos baseados na Densidade

Os algoritmos que adoptam métodos baseados na densidade têm como ideia chave o seguinte: identificar grupos como regiões densas de objectos e considerar como ruído regiões com baixa densidade de objectos. Deste tipo de algoritmos destaca-se o algoritmo DBSCAN [**21**], bastante referenciado na literatura, analisado de seguida.

DBSCAN

Antes de mais, este algoritmo requer dois parâmetros: o $epsilon$ (eps), que denota o raio máximo da vizinhança de um ponto; $MinPts$, que representa o número mínimo de pontos que têm de estar na sua vizinhança para verificar a condição de *core point*. O conjunto de definições em que o algoritmo se baseia é o seguinte:

- ***ε-Neighborhood*:** Define o conjunto de objectos que estão a uma distância $\leq eps$ de um ponto p. Este conjunto é representado com a seguinte notação: $N_{Eps}(p)$;
- ***Directly density-reachable*:** Um ponto p é considerado *directly density-reachable* de q caso verifique as seguintes condições:
  - $p \in N_{Eps}(q)$;
  - $|N_{Eps}(q)| \geq MinPts$, condição de *core point*.
- ***Density-reachable:*** Um ponto p é *density-reachable* de q caso exista uma cadeia de pontos $p_1, \ldots, p_n$, $p_1 = q, p_n = p$ tal que p_{i+1} é *directly density-reachable* de p_i;
- ***Density-connected*:** Um ponto p é *density-connected* de q se existir um ponto o tal que ambos os pontos p e q são *density-reachable* de o.
- ***Cluster (Grupo)*:** Qualquer grupo C identificado pelo algoritmo tem de verificar as seguintes condições:
  - $\forall\, p, q: Se\ p \in C\ e\ q$ é $density\text{-}reachable\ de\ p\ então\ q\ \in C$
  - $\forall\, p, q \in C: p$ é $density\text{-}connected\ a\ q$
- ***Noise*:** Todos os pontos que não façam parte de um grupo.

A Figura 19, obtida de [**16**], ilustra os conceitos *density-reachable* e d*ensity-connected.*

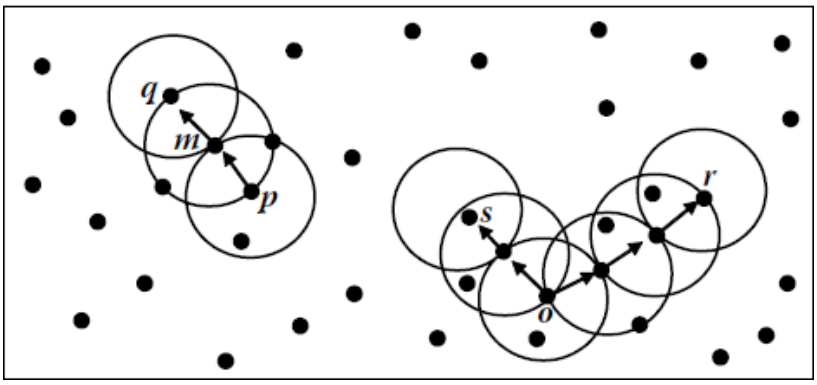


Figura 19 - Conceitos: a) q é density-reachable de p; b) s é density-connected de r.

O algoritmo inicia com um ponto arbitrário p não visitado. Obtém todos os pontos *density-reachable* do ponto p. Caso o ponto verifique a condição de *core point* é criado um grupo, caso contrário, não é criado qualquer grupo. Em qualquer situação o ponto é identificado como visitado. O processo termina quando não existem objectos que possam ser adicionados a qualquer grupo.

A complexidade do algoritmo depende do modo como se obtêm todos os pontos *density-reachable*. Na presença de uma base de dados em que seja utilizado um índice espacial a complexidade temporal é $O(n * \log(n))$, caso contrário a complexidade é $O(n^2)$, em que n representa a dimensão do conjunto de dados. Deste modo, apresenta um bom potencial de escalabilidade. De notar que este algoritmo preenche a maior parte dos requisitos enunciados anteriormente. Contudo, peca por se basear numa densidade global, isto é, os parâmetros de entrada traduzem a densidade mínima necessária para a formação de um grupo. Caso existam grupos com diferentes densidades, o resultado do algoritmo não será o ideal. Apesar da existência de parâmetros de entrada, os autores deste algoritmo definem uma heurística para colmatar esse aspecto negativo.

Ao longo dos últimos anos, foram propostas melhorias para este algoritmo. Com o objectivo de corrigir algumas falhas foi proposto ST-DBSCAN [**22**], como também foram vários os trabalhos para melhorar os tempos de resposta, designadamente IDBSCAN [**23**].

2.3.2.4 Métodos baseados em Grelha

Em geral, os métodos baseados em grelha transformam o espaço original do conjunto de dados num determinado número de células, criando uma estrutura em grelha. Posteriormente, é aplicado um algoritmo sobre o espaço transformado.

WaveCluster

Considere-se o domínio inicial $S = A_1 * A_2 * \dots * A_d$, ($A_i$ representa uma dimensão do domínio inicial) onde d é a dimensionalidade do espaço e O o conjunto de pontos com dimensionalidade d, $O = \{O_1, O_2, \dots, O_n\}$, com $O_i = (O_{i1}, O_{i2}, \dots, O_{id})$. Inicialmente, o domínio original é dividido em hiper-rectângulos não sobrepostos, designados de *células*. Estas são obtidas através da segmentação de cada dimensão A_i, em m_i intervalos. Cada célula é a intersecção de um intervalo em cada dimensão. Em cada célula é guardado o número de pontos que estão contidos dentro da célula (contador). A grelha resultante é designada como *espaço quantizado*.

Á semelhança do algoritmo DBSCAN, o algoritmo WaveCluster [**24**] também se baseia num conjunto de definições, apresentadas abaixo:

- **Célula Vazia:** É uma célula no espaço transformado com contador igual a zero;
- **Célula Não Vazia:** Define uma célula com contador maior que zero;
- **Célula Significante:** Designa uma célula com contador acima de um determinado limiar T;
- ***ε-Neighborhood*:** Uma célula c_1 é *ε-Neighborhood* de uma célula c_2 caso ambas sejam células significantes, não vazias e com $d(c1, c2) < \varepsilon$, onde d denota a distância entre as duas células e ε > 0;
- ***k-ε-Neighborhood*:** Uma célula c_1 é *k-ε-Neighborhood* de uma célula c_2 se ambas forem significantes, não vazias e c_1 é um dos k vizinhos (ε-Neighborhood) de c_2;
- ***k-connected*:** Duas células c_1 e c_2 são *k-connected* se existe uma sequência de células $p_1, p_2, \dots, p_j$ tal que $p_1 = c_1$, $p_j = c_2$ e $p_i + 1$ é *k-ε-Neighborhood* de p_i, em que $1 \leq i < j$;
- **Grupo:** Um grupo C é definido por um conjunto de células significantes $\{c_1, c_2, \dots, c_n\}$ que se encontrem *k-connected* no espaço transformado.

Deste modo, após *WaveCluster quantizar* o espaço original, o objectivo do algoritmo é encontrar os grupos de acordo com a definição acima. Assim, é aplicada uma transformação *wavelet* ao espaço *quantizado*, de modo a converter este espaço para o domínio da frequência. É observado o espaço original de uma perspectiva de processamento de sinal: altas frequências correspondem às fronteiras dos grupos; baixas frequências com altas amplitudes correspondem a zonas, no espaço original, onde existe uma grande quantidade de objectos (grupos).

Considerando conjuntos de dados com dimensionalidade $d = 2$, a complexidade temporal do algoritmo é $O(n)$, onde n é a dimensão do conjunto de objectos. Dos algoritmos estudados, é o mais escalável. Além disso, preenche globalmente os requisitos enunciados anteriormente: não é afectado por *outliers*; não é sensível à ordem de entrada dos dados; não é necessário o conhecimento *à priori* do número de grupos e é capaz de encontrar grupos com formas arbitrárias. Ainda assim, mantém a necessidade de parâmetros de entrada. Os principais são: (*i*) a resolução da grelha para cada dimensão (m_i); (*ii*) o valor do limiar T.

No entanto, em [**24**] é feita a seguinte afirmação: "*When the number of objects is high, we can apply signal-processing techniques(...)*". Num contexto SOLAP, é necessário considerar que, numa interrogação, nem sempre irá resultar um número muito elevado de valores e, ainda assim, se pode verificar a desorganização no mapa. Deste modo, apesar de ser um algoritmo potencialmente bom, não é aplicável para o objectivo pretendido.

2.3.3 Aplicações: Agrupamento de Pontos

Existem ferramentas que têm como objectivo melhorar e estender a *API* da *Google Maps*. Não só têm como objectivo aumentar os níveis de desempenho [**25**] que a *API* oferece, como também possibilitam uma melhor organização do mapa. Assim, as ferramentas que têm como objectivo estender a *API* da *Google Maps*, e que introduzem algoritmos de agrupamento de pontos são: (*i*) *MarkerCluster* [**26**]; (*ii*) *ClusterMarker* [**27**].

Em [**28**] foi implementado um protótipo que tem como finalidade remover regiões com uma grande densidade de pontos (marcadores).

Outra abordagem existente que difere um pouco das anteriores designa-se *regional clustering* [**29**], e consiste em agrupar marcadores que pertencem a uma designada "região". Como "região" pode-se considerar cidades, freguesias, concelhos, distritos, entre outras.

2.3.3.1 MarkerCluster

MarkerCluster é uma solução baseada em grelha. Esta ferramenta irá agrupar marcadores de acordo com a sua distância a um centro de um grupo. Quando o marcador é adicionado, este irá pesquisar a sua posição em todos os grupos. Caso não seja colocado em nenhum grupo, será criado um novo grupo com o marcador.

Cada grupo criado é representado por um marcador que contém o número de elementos do grupo, como ilustra a Figura 20, retirada de [**25**].

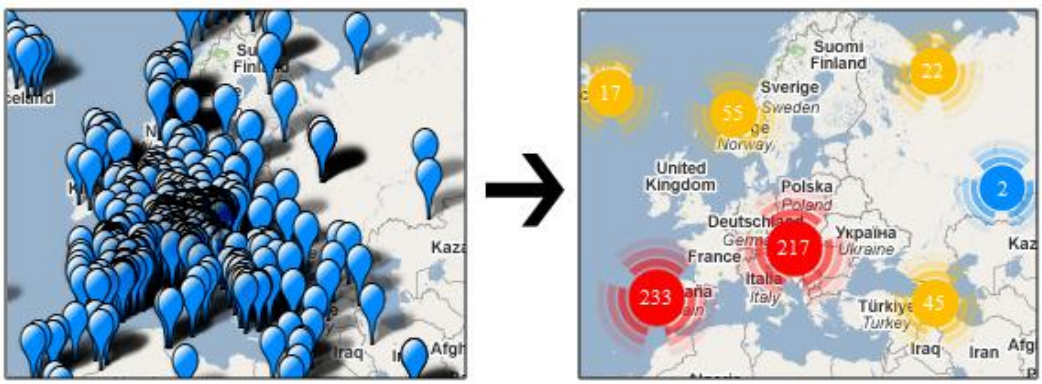


Figura 20 - Exemplo da Utilização da API MarkerCluster, obtida de [25].

Os parâmetros de entrada do algoritmo são: (*i*) tamanho da célula; (*ii*) nível máximo de zoom. O primeiro designa o tamanho de cada célula em pixéis; o segundo denota o nível máximo de zoom monitorizado pela *API*, isto é, quando o mapa se encontra para lá do nível máximo de zoom, a *API* deixa de aplicar o algoritmo de agrupamento.

A complexidade do algoritmo é linear $O(k)$, onde k representa o número de grupos existentes antes de adicionar o novo marcador. No entanto, o tempo de execução depende do parâmetro *tamanho da célula*. Este algoritmo depende da ordem de entrada dos dados. Poderá levar a grupos que, muito provavelmente, não seriam considerados por um comum utilizador. Considerando a situação da Figura 21, em a) encontra-se a situação inicial, em b) encontra-se o resultado do algoritmo e em c) encontra-se o resultado que, provavelmente, um utilizador considerava. Também não possibilita encontrar grupos com formas arbitrárias devido à noção de célula em que se baseia.

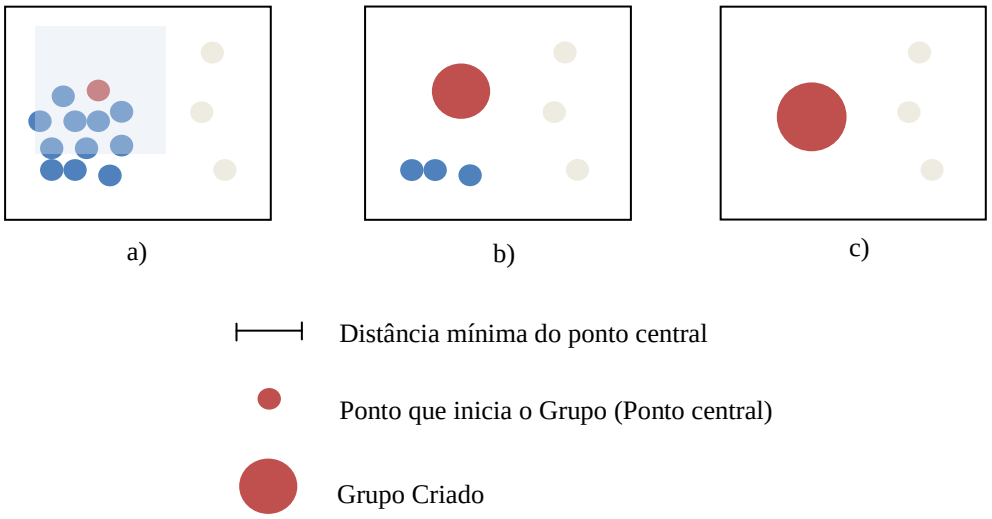


Figura 21 - Exemplo da uma possível situação da execução da API MarkerCluster.

2.3.3.2 ClusterMarker

ClusterMarker é uma *API* cujo algoritmo se baseia numa noção de intersecção. *ClusterMarker* detecta qualquer grupo de dois ou mais marcadores que se intersectam no mapa. Cada grupo de marcadores é representado por um único marcador. A Figura 22 mostra um exemplo da utilização desta *API*.

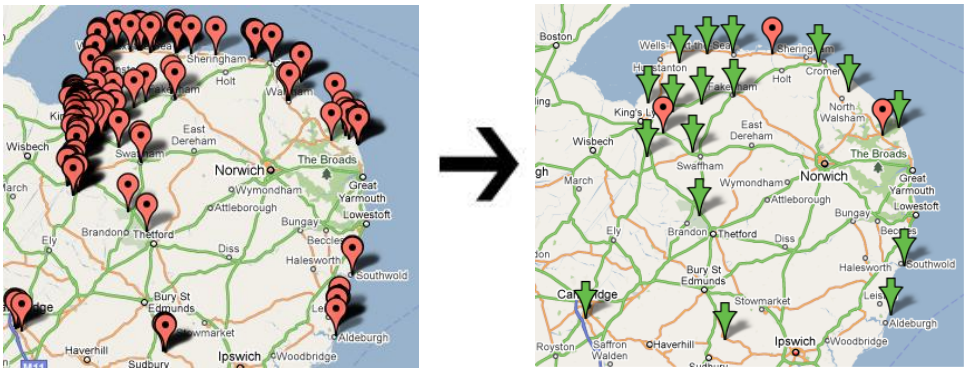


Figura 22 - Exemplo da Utilização da API ClusterMarker.

O algoritmo concebido pela ferramenta *ClusterMarker* não contém qualquer parâmetro de entrada. Esta característica é bastante importante, porque permite uma maior automatização do processo de agrupamento. A sua complexidade é quadrática $O(n^2)$, onde n representa o número de marcadores visíveis no *canvas*.

Considerar apenas a intersecção de marcadores pode, eventualmente, levar a resultados de baixa qualidade, como pode ser observado na Figura 23. Neste contexto, o algoritmo retorna b), apesar do utilizador potencialmente preferir o resultado de c). Esta situação não se verifica no exemplo da Figura 22 pois na *API* em cada iteração são "escondidos" os marcadores agrupados.

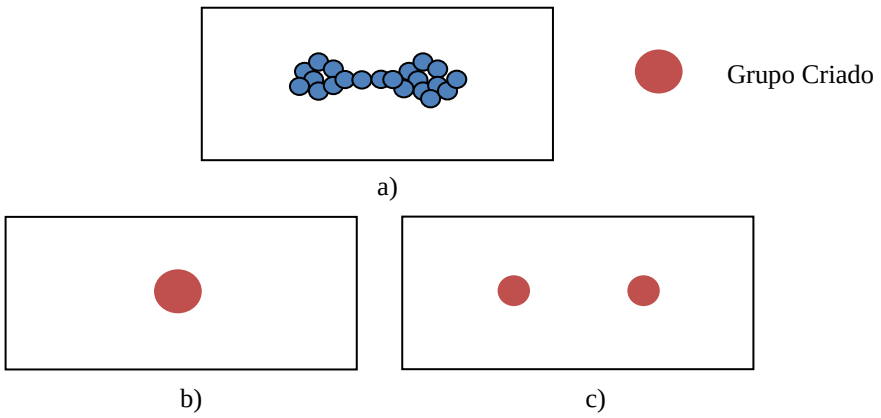


Figura 23 - Exemplo da uma possível situação da execução da *API ClusterMarker*.

2.3.3.3 Protótipo desenvolvido por Wannes Meert

O protótipo desenvolvido está estruturado segundo uma arquitectura *3-tier* (nível de interface, nível algorítmico, nível de dados). O processo de agrupamento está dividido em quatro passos, em que todos os níveis cooperam para agrupar os marcadores.

O primeiro passo consiste num pedido inicial, por parte do nível de interface em que, nesse pedido, são enviadas as coordenadas que delimitam o *canvas*. De seguida, são utilizadas estas coordenadas para efectuar uma interrogação à base de dados (nível dos dados). Do resultado da interrogação esperam-se todos os pontos contidos no *canvas*. Em terceiro lugar, é aplicado o algoritmo DBSCAN, já referido anteriormente. Por último, é gerado um ficheiro XML com o resultado a partir do qual são gerados os marcadores.

De notar que as coordenadas guardadas na base de dados para cada ponto são a longitude e a latitude. Ao contrário das soluções anteriores, está-se a trabalhar ao nível da representação geográfica (longitude, latitude) e não sobre a projecção geográfica do mundo real (*canvas*). A acrescentar que o algoritmo DBSCAN é aplicado com os parâmetros definidos pelo utilizador e não é utilizada qualquer heurística. Na Figura 24 é ilustrado um exemplo da utilização deste protótipo.

Figura 24 - Exemplo da uma execução da utilização do Protótipo.

Como pode ser observado, cada grupo é representado por um rectângulo que corresponde à área onde se situam os marcadores do respectivo grupo. Na realidade, esta solução mostra-se deselegante, uma vez que: (*i*) faz com que a representação de um grupo ocupe demasiado espaço no mapa, mantendo o mapa preenchido com objectos geográficos; (*ii*) pode levar à sobreposição das representações, o que não é desejável.

2.3.3.4 Regional Clustering: Travellr

Em linhas gerais, esta solução apresenta um algoritmo que agrupa os marcadores com base numa noção de região, isto é, em conformidade com o nível de zoom é escolhido por que tipo de região são agrupados os marcadores, como pode ser observado na Figura 25, obtida de [**30**].

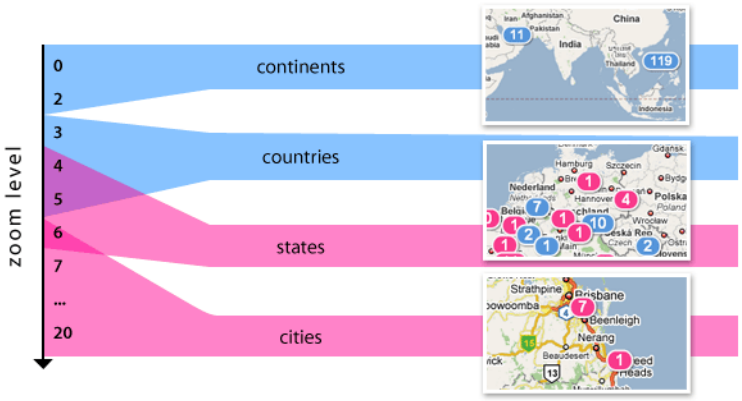


Figura 25 - Travellr: Detalhes de Implementação, obtida de [30**].**

O algoritmo implementado por esta aplicação tem complexidade temporal $O(n * m * k)$, onde n é o número de pontos que é necessário agrupar, m é o número distinto de regiões visíveis no mapa, pelas quais se está a efectuar o agrupamento, e k é o número distinto de tipos de regiões pelas quais se está a realizar o agrupamento. O número de grupos corresponde a $m * k$.

O algoritmo, ao agrupar pontos considerando dois ou mais níveis de regiões ($k > 1$), poderá manter o mapa com muitos pontos e de difícil leitura, como é possível observar no mapa intermédio da Figura 25.

2.3.4 Conclusão sobre o Agrupamento Espacial de Pontos

Ao longo das secções 2.3.2 e 2.3.3 foram discutidas soluções tanto académicas como comerciais para realizar o agrupamento de pontos com base na proximidade destes. Antes de ser tomada qualquer escolha sobre o algoritmo que irá ser integrado no modelo genérico SOLAP [**1**], é necessário realizar uma síntese dos algoritmos estudados. Deste modo, na tabela abaixo é apresentado um resumo das características dos algoritmos com base no que foi discutido anteriormente e em estudos já realizados [**16**] [**15**] [**14**] [**17**]. A escala da classificação usada tem quatro valores representativos, indo de pior para o melhor, pelos símbolos --, -, +, ++. Por exemplo, algoritmos com complexidade linear são classificados com ++ e os algoritmos com complexidade quadrática são classificados com +.

	Académicas						Comerciais	
	k-Means	k-Medoid	CLARANS	Chameleon	DBSCAN	WaveCluster	MarkerCluster	ClusterMarker
Escalabilidade	-	- -	- -	+	+/++	++	++	+
Versatilidade	- -	- -	++	-	++	-	- -	- -
Formas arbitrárias	- -	- -	- -	++	+/++	++	- -	-/+
Mínimo Conhecimento do domínio	- -	- -	-	-	+	+	-	++
Robusto na presença de ruído	- -	-	-	-/+	++	++	++	++
Insensível à ordem de entrada	- -	- -	-	++	++	++	- -	- -

Tabela 2 - Algoritmos Versus Requisitos.

Após a análise de diferentes algoritmos, é observável que os algoritmos DBSCAN e WaveCluster são aqueles que apresentam as melhores características. No entanto, o algoritmo WaveCluster não é aplicável para o objectivo pretendido, como foi analisado anteriormente. Assim, para o objectivo de agrupamento de pontos, num contexto SOLAP, o algoritmo DBSCAN apresenta-se como a melhor escolha dos algoritmos estudados.

A aproximação *regional clustering* é diferente de todas as outras e com objectivos diferentes. Devido a esse factor não foi efectuada uma análise comparativa com as outras soluções. Porém, é uma aproximação interessante e que será integrada num contexto SOLAP, através da utilização das hierarquias espaciais. O modo como é definida e integrada a solução com base na ideia chave de *regional-clustering*, no modelo genérico SOLAP, será apresentado no Capítulo 3.

2.3.5 Algoritmos: Agrupamento de Polígonos

Os polígonos são formas dispersas no espaço. Aos requisitos identificados em 2.3.1 é necessário introduzir outros factores: (*i*) topologia; (*ii*) direcção; (*iii*) quantidade de fronteira partilhada entre os polígonos.

Actualmente, a maior parte dos algoritmos de agrupamento espacial está apenas focada em agrupar conjuntos de pontos. Porém, existe a necessidade de agrupar outras formas geométricas,

nomeadamente em ambientes SOLAP. Uma aplicação importante, neste tipo de ambientes, designa-se de *regionalização*. Este processo consiste em agrupar diversas regiões com base num critério de proximidade entre as regiões.

2.3.5.1 GDBSCAN

O algoritmo GDBSCAN [**31**] consiste numa visão generalizada do algoritmo DBSCAN, permitindo o agrupamento de qualquer objecto. Em vez de existir uma noção de vizinhança de pontos (*ε-Neighborhood*), utiliza-se uma noção de vizinhança com base num predicado binário que é simétrico e reflexivo, designado de *NPred*. Não conta simplesmente o número de objectos na vizinhança, mas utiliza outras métricas para determinar a cardinalidade da vizinhança, denominada de *wCard*. Nesta perspectiva, é possível aplicar este algoritmo no agrupamento de polígonos. Uma solução proposta pelos autores é considerar *NPred* a intersecção de polígonos e *wCard* como o somatório das áreas intersectadas.

2.3.5.2 P-DBSCAN

O algoritmo P-DBSCAN [**32**] é baseado no algoritmo DBSCAN. Algumas definições em que se baseia o DBSCAN são redefinidas de modo a se adaptarem ao agrupamento de polígonos. Essas definições são as seguintes:

- ***ε-Neighborhood*:** A vizinhança de um polígono p, designada por $N_\varepsilon(p)$, é definida como $N_\varepsilon(p) = \{ q \in D \mid dist(p,q) \leq \varepsilon\}$, onde D representa o conjunto de dados e $dist(p,q)$ designa a distância entre um polígono p e q;
- ***Core polygon*:** Um polígono p é *core polygon* se contiver pelo menos *MinPolys* polígonos, dentro da sua vizinhança, e pelo menos *MinS* partições radiais espaciais não vazias, isto é, $\left[Count_{i=1}^{R}\left(N_{\varepsilon,i}(p) \neq 0\right)\right] \geq MinS$;
- ***Border polygon*:** Esta definição é semelhante à anterior, no entanto, apenas tem que verificar a seguinte condição:$\left[Count_{i=1}^{R}\left(N_{\varepsilon,i}(p) \neq 0\right)\right] > R - MinS$;
- ***Outlier polygon*:** Um polígono p é considerado ruído quando, a uma distância ε, não está presente qualquer polígono.

As restantes definições, *directly-density-reachable*, *density-reachable*, *density-connected*, e *cluster* são derivações directas das definições do algoritmo DBSCAN.

Como já foi referido, é necessário ter em conta as propriedades topológicas dos polígonos. Com este intuito, os autores introduziram a definição de ***radial spatial neighborhood*** de um polígono. A

definição dita que a vizinhança de um polígono p pode ser particionada. Assim, $N_\varepsilon(p) = \bigcup_{i=1}^{R} N_{\varepsilon,i}(p)$, em que R é o número de sectores de igual tamanho que divide radialmente o espaço em redor do polígono p.

A métrica de similaridade utilizada para o cálculo da distância entre polígonos é a distância de *hausdorff* ajustada, que será apresentada na secção 2.3.6.2. O algoritmo P-DBSCAN é semelhante ao algoritmo DBSCAN.

2.3.6 Funções de Similaridade: Agrupamento de Polígonos

Outra forma de lidar com o agrupamento de polígonos é utilizar os algoritmos já existentes para o agrupamento de pontos, mas com novas funções de similaridade. Em vez de calcular a distância euclidiana, ou outra qualquer distância focada para o cálculo da distância entre pontos, utiliza funções de similaridade desenvolvidas para calcular distância entre polígonos.

2.3.6.1 CLARANS

Os autores de CLARANS desenvolveram três novas funções de similaridade [**19**] para lidar com o agrupamento de polígonos convexos, tais como: (*i*) distância de separação exacta; (*ii*) distância mínima entre vértices; (*iii*) distância entre rectângulos isotéticos.

Distância de Separação Exacta

Dados dois polígonos A e B, a distância é definida como a distância mínima entre quaisquer pares de pontos P e Q, isto é, $min\{d(P,Q) \mid P \in A, Q \in B\}$ cujo P e Q são pontos de A e B respectivamente (Figura 26). A esta distância designa-se *distância de separação exacta*. O seu cálculo, entre dois polígonos, é obtido em dois passos: primeiro verifica se os dois polígonos se intersectam e, caso se confirme, o valor da distância é zero. Caso contrário, calcula a distância de separação entre os dois polígonos. O algoritmo utilizado para realizar o cálculo da *separação exacta* implementado em [**19**] tem complexidade $O(log(n) + log(m))$, onde n e m representam os números de vértices dos polígonos A e B, respectivamente. A complexidade agregada das duas operações mantém-se igual à anterior.

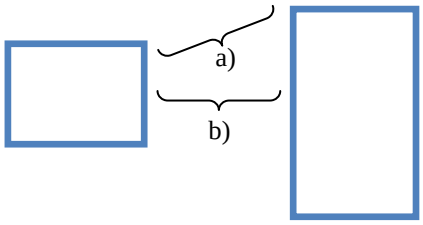
Distância Mínima entre Vértices

Outro modo de calcular a distância entre dois polígonos é calcular a distância mínima entre os seus vértices, isto é, $min\{d(P,Q) \mid P \in A, Q \in B\}$ cujo P e Q são vértices de A e B, respectivamente (ex: Figura 26). A complexidade desta aproximação é $O(n * m)$. Apesar de a

complexidade ser superior à complexidade da aproximação anterior, tipicamente mostra melhores tempos de execução quando n e m são valores pequenos, isto é, menores ou iguais a vinte.

Distância entre Rectângulos Isotéticos

Outra abordagem é computar os dois rectângulos isotéticos[3], dos polígonos A e B, e calcular a distância de separação exacta entre estes rectângulos. A complexidade para calcular os rectângulos isotéticos é $O(n)$ e a complexidade para calcular a distância entre os rectângulos isotéticos é $O(1)$. Esta forma de distância permite que o algoritmo CLARANS seja aplicável a qualquer polígono e não só para os convexos, ao contrário das aproximações anteriores. Assim, apresenta-se uma solução mais eficiente.



a) Distância Mínima entre Vértices;

b) Distancia entre Rectângulos Isotéticos ou Distância de Separação Exacta

Figura 26 - Ilustra distância entre polígonos.

2.3.6.2 Métrica de Dissimilaridade

Dados dois polígonos P_i e P_j, a função de dissimilaridade proposta por [**33**] é a seguinte:

$$D(P_i, P_j) = f\left(D_{ns}(P_i, P_j), D_s(P_i, P_j)\right),$$

onde a função D_{ns} denota a função que calcula a dissimilaridade entre dois polígonos com base nos seus atributos não espaciais, e D_s calcula a dissimilaridade entre dois polígonos com base nos seus atributos espaciais. Como o âmbito desta dissertação é agrupar polígonos com base nos seus atributos espaciais, irá ser detalhada a função D_s e a forma como esta foi concebida.

É necessário apresentar a terminologia que é utilizada pelos autores. Os atributos espaciais de um polígono podem ser divididos em duas categorias: (*i*) intrínsecos; (*ii*) extrínsecos. Os primeiros são propriedades relativas a um polígono, tais como localização, forma, área, entre outras. Os atributos extrínsecos abrangem os vários objectos espaciais que possam coexistir dentro de um ou mais polígonos. Existem três classes de objectos que são considerados: pontos, linhas e objectos poligonais. Desta forma, a função de dissimilaridade D_s é decomposta em:

[3] Dado um polígono A, o rectângulo isotético I_a corresponde ao menor rectângulo que contém A, e as respectivas arestas são paralelas tanto ao eixo das abcissas como ao eixo das ordenadas.

$$D_s(P_i, P_j) = d_{ins}(P_i, P_j) + d_{ext}(P_i, P_j),$$

em que d_{ins} denota a função de dissimilaridade entre atributos intrínsecos e d_{ext} designa a dissimilaridade entre atributos extrínsecos. Assim, d_{ins} corresponde:

$$d_{ins}(P_i, P_j) = d_{hs}(P_i, P_j) + \sqrt{\sum_{k=1}^{m}(t_{ik} - t_{jk})^2},$$

onde t_k representa o *k-ésimo* atributo intrínseco espacial do polígono *i* ou *j*. Finalmente d_{hs} consiste na distância de *hausdorff* ajustada proposta em [**32**] que consiste:

$$d_{hs}(P_i, P_j) = \left(1 - \frac{2 * S_{ij}}{S_i + S_j}\right) * d_h(p_i, p_j),$$

onde d_h denota a distância original de hausdorff, S_i e S_j representam os perímetros de cada polígono e S_{ij} consiste na quantidade de fronteira partilhada pelos polígonos. A função d_{ext} é definida da seguinte forma:

$$d_{ext}(P_i, P_j) = w_\tau d_\tau(P_i, P_j) + w_\varphi d_\varphi(P_i, P_j) + w_\omega d_\omega(P_i, P_j).$$

De modo a medir a distância entre dois polígonos com base nos atributos extrínsecos, é necessário ter em mente diversos atributos que possam ter os objectos espaciais: (*i*) extensão; (*ii*) densidade; (*iii*) distribuição; (*iv*) topologia; (*v*) direcção. Devido às diferenças entre os objectos que possam estar contidos nos polígonos (pontos, linhas, polígonos) existe a possibilidade de atribuir diferentes pesos a cada tipo de objecto. De notar que a soma dos pesos tem de verificar a seguinte condição: $w_\tau + w_\varphi + w_\omega = 1$.

Ao contrário das funções de similaridade analisadas na secção 2.3.6.1, esta métrica inclui diversos factores que podem e devem influenciar a dissimilaridade entre polígonos, em particular a quantidade de fronteira partilhada entre polígonos.

2.3.7 Conclusão sobre o Agrupamento Espacial de Polígonos

Ao longo das secções 2.3.5 e 2.3.6 foram discutidas possíveis soluções para abordar o problema de agrupamento de polígonos. Após esta análise, a solução que nos parece mais adequada consiste no algoritmo P-DBSCAN. Potencialmente poderá ser utilizada a função de dissimilaridade apresentada na secção 2.3.6.2 com o algoritmo P-DBSCAN.

Capítulo 3

Extensão ao SOLAP+

Este capítulo apresenta as propostas para a extensão do modelo genérico SOLAP, apresentando inicialmente os conceitos em que se baseia o desenvolvimento do sistema SOLAP+.

Ao longo deste capítulo são apresentadas propostas para a extensão do modelo genérico desenvolvido previamente em [**1**]. Estas têm o propósito de permitir analisar, em simultâneo, dois atributos espaciais de diferentes dimensões, integrar algoritmos de agrupamento espacial e incorporar um gestor de estilos no processo de visualização. Inicialmente são apresentados os conceitos e os mecanismos em que se baseia o desenvolvimento deste sistema SOLAP+.

3.1 Sistema SOLAP+

O sistema SOLAP+ foi desenvolvido por Ruben Jorge [**1**]. A definição do modelo genérico, em que se baseia o sistema, tem por base o seguinte conjunto de definições:

- **Atributo Semântico (aS):** representa um atributo do tipo alfanumérico presente numa dimensão;
- **Atributo Espacial (aEP):** designa um atributo de uma dimensão que contém informação quanto à coordenada ou conjunto de coordenadas do objecto espacial que representa. Um atributo semântico associado ao atributo espacial tem a seguinte notação: **aS(aEP)**. O modelo pressupõem sempre a existência de um **aS(aEP)** pelo facto de um atributo espacial estar associado a uma descrição alfanumérica (ex: aS(aEP) descreve o nome dos países e aEP contém a descrição do polígono de cada país).
- **Métrica Numérica (mN):** denota uma métrica numérica associada a uma tabela de facto.

No modelo são considerados dois tipos de dimensões: semântica ou espacial. Basta que uma dimensão contenha pelo menos um atributo espacial para que esta seja definida como dimensão espacial.

Em função do pretendido pelo utilizador, o conjunto de dados é obtido através da execução de interrogações sobre a base de dados. Como podemos observar através da Figura 27 obtida de [**1**], neste modelo genérico uma interrogação está sujeita a dois fluxos distintos. Um primeiro fluxo tem como objectivo apresentar os dados ao nível do mapa e correspondente nível da tabela de suporte, e um segundo fluxo que é responsável por enviar dados para o nível de detalhe. Quanto ao primeiro fluxo, está sujeito a diversas fases intermédias: (*i*) pré-processamento; (*ii*) particionamento; (*iii*) vectorização.

No modelo actual, a fase de pré-processamento é uma fase ainda não explorada e a sua aplicação é facultativa.

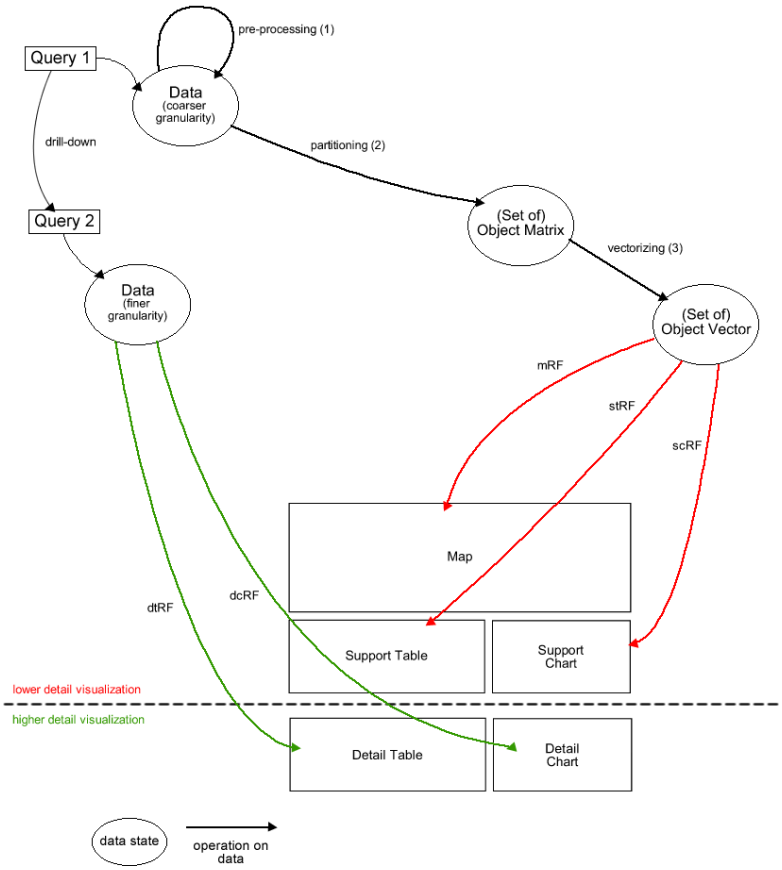


Figura 27 - Processamento de Interrogações.

Particionamento é uma fase que tem como objectivo particionar em n grupos o conjunto de linhas inicial, onde cada grupo corresponde a uma das combinações entre os atributos espaciais resultantes da interrogação (Figura 28, obtida de [**1**]).

spA 1	spA 2	sA	M1	M2
□	○	B	3	5
□	○	C	7	10
□	△	A	10	20
□	△	B	10	15
□	△	C	5	10
○	□	A	12	10
△	○	C	2	8

Figura 28 - Fase de particionamento.

Para cada grupo é aplicada a fase de vectorização que converte as diferentes linhas num único vector, designado de vector de objectos ($vObj$), como pode ser observado na Figura 29, retirada de [**1**].

Figura 29 - Fase de vectorização.

Após todas as transformações a que está sujeito o conjunto de linhas inicial, obtém-se o conjunto de vectores com os dados devidamente transformados. Posteriormente, consoante a que componente se destina, o conjunto de vectores está sujeito a diferentes funções de representação:

- Função de representação para a zona do mapa (mRF): retorna os objectos espaciais no mapa;
- Função de representação para a tabela de suporte ($stRF$): o resultado de saída corresponde à tabela de suporte;
- Função de representação para o gráfico de suporte ($scRF$): retorna o gráfico de suporte;
- Função de representação para a tabela de detalhe e respectivo gráfico de detalhe ($dtRF$ e $dcRF$) são semelhantes relativamente ao nível de suporte.

Após a definição da estrutura base, o autor definiu um modelo de interacção genérico que previa os seguintes casos de interacção:

- Cenário base: Um atributo espacial e correspondente atributo semântico:
  - Caso 1: Uma métrica numérica;
  - Caso 2: Múltiplas Métricas Numéricas;
- Cenário base: Uma ou mais métricas numéricas em adição ao atributo espacial e correspondente atributo semântico:
  - Caso 3: Atributos semânticos de dimensões semânticas;
  - Caso 4: Atributos semânticos de dimensões espaciais;
  - Caso 5: Atributos espaciais da mesma dimensão.

Em todos os casos de interacção verifica-se a relação de 1:1 entre o mapa e a tabela de suporte. Para tal, a função de representação da tabela de suporte ($stRF$) retorna uma tabela pivô sujeita a determinadas restrições. A organização do cabeçalho desta tabela é a seguinte (Figura 30):

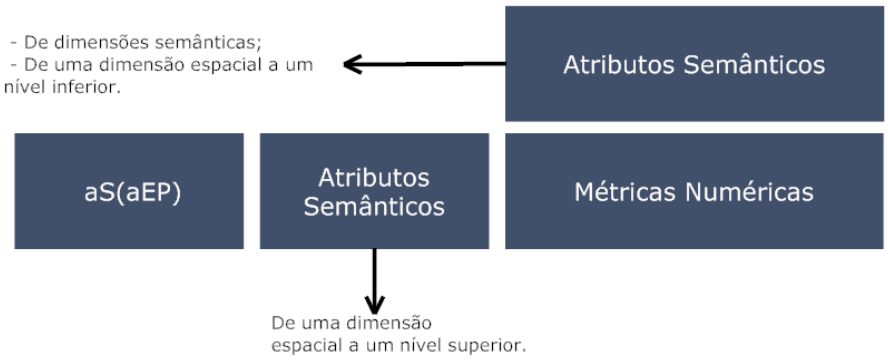


Figura 30 - Estrutura da Tabela de Suporte.

No lado esquerdo encontram-se os atributos semânticos associados aos atributos espaciais. Existem duas zonas para os atributos semânticos. A zona que é utilizada por um determinado atributo semântico depende de alguns factores. Existe também uma área para as métricas numéricas.

Quando se está na presença de atributos semânticos haverá potencialmente vários valores destes para cada valor do atributo aS(aEP), o que provocaria várias linhas dado um valor de aS(aEP) e impossibilitaria manter a relação de 1:1 entre a tabela de suporte e o mapa. Portanto, os valores dos atributos semânticos têm que ser colocados como cabeçalhos quando provêm:

1. de dimensões semânticas;
2. de dimensões espaciais mas a um nível inferior de granularidade comparativamente com aS(aEP).

Recorde o exemplo do qual resultou a tabela seguinte (Figura 31):

	Semestre	
	Primeiro	Segundo
Nome do Hipermercado	**Total de Vendas**	**Total de Vendas**
X	1.000.000.000	3.000.000.000

Figura 31 - Exemplo de tabela de suporte.

Se esta tabela não tivesse respeitado a *restrição 1*, então para o hipermercado X haveria duas linhas, e neste caso a relação de 1:1 entre a tabela de suporte e um possível mapa não se verificaria.

As restantes secções referem-se exclusivamente às propostas desta dissertação para a extensão do modelo genérico desenvolvido previamente em [**1**].

3.2 Caso 6: Dois Atributos Espaciais de Diferentes Dimensões

Este caso de interacção é caracterizado pela utilização de dois atributos espaciais e uma ou várias métricas numéricas. O vector de objectos representativo é definido por:

$$vObj = \{aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2), mN_1, \dots, mN_n\}.$$

Ao longo desta secção, é assumido que o atributo espacial aEP_1 é aquele que se verifica já em análise, enquanto que o atributo espacial aEP_2 é posteriormente adicionado. Além disso, é utilizado um conjunto de dados num domínio genérico representado sob a forma $\mathcal{D}(aS_1(aEP_1)) = \{A, B, C, D, E\}$ e $\mathcal{D}(aS_2(aEP_2)) = \{1,2,3,4\}$.

Como já foi referido, a propriedade 1:1 entre a tabela de suporte e o mapa é fundamental no modelo genérico SOLAP [**1**]. Deste modo, considerando os possíveis tipos de dados geométricos dos atributos espaciais, as situações que podem ocorrer são:

- Situação 1: aEP_1: Ponto e aEP_2: Ponto;
- Situação 2: aEP_1: Ponto e aEP_2: Polígono ou vice-versa;
- Situação 3: aEP_1: Polígono e aEP_2: Polígono;
- Situação 4: aEP_1: Ponto e aEP_2: Linha ou vice-versa;
- Situação 5: aEP_1: Polígono e aEP_2: Linha ou vice-versa;
- Situação 6: aEP_1: Linha e aEP_2: Linha.

3.2.1 Ponto com Ponto

A primeira situação verifica-se quando ambos os objectos geográficos dos atributos em análise são pontos. O retorno da função $mRF = f(aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2), mN_1)$ representa cada relação entre o atributo $aS_1(aEP_1)$ e o atributo $aS_2(aEP_2)$ através de uma linha orientada que conecta ambos os objectos espaciais, como pode ser observado na Figura 32.

$aS_1(aEP_1)$	$aS_2(aEP_2)$	mN_1
A	1	4
B	2	10
C	3	20
D	4	7
D	3	19
E	2	2

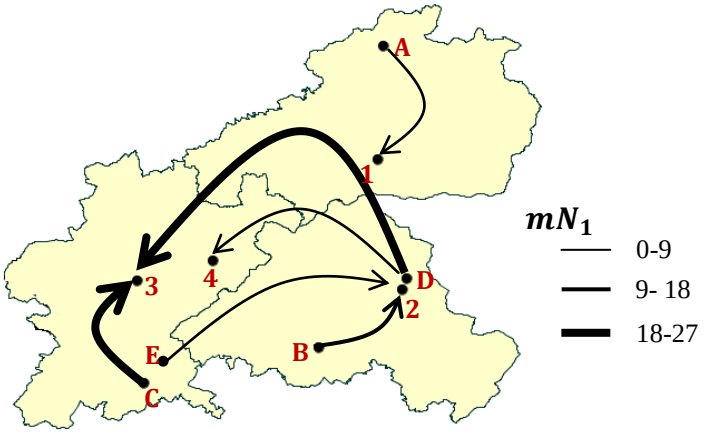


Figura 32 - Tabela de Suporte e respectivo resultado da função *mRF.*

A espessura da linha, que conecta ambos os objectos espaciais, é dada pelo valor da métrica (a legenda tem implícita a notação maior ou igual a *X* e menor que *Y*). Através desta solução é possível obter uma visão global das relações entre os atributos espaciais, num único mapa, permitindo realizar análises comparativas entre eles. Sem esta solução, no sistema anterior, teria que se realizar *slices* num dos atributos espaciais, e com isso obter-se-iam diversos mapas, como é ilustrado na Figura 33 (*slices* realizados sobre o atributo $aS_2(aEP_2)$).

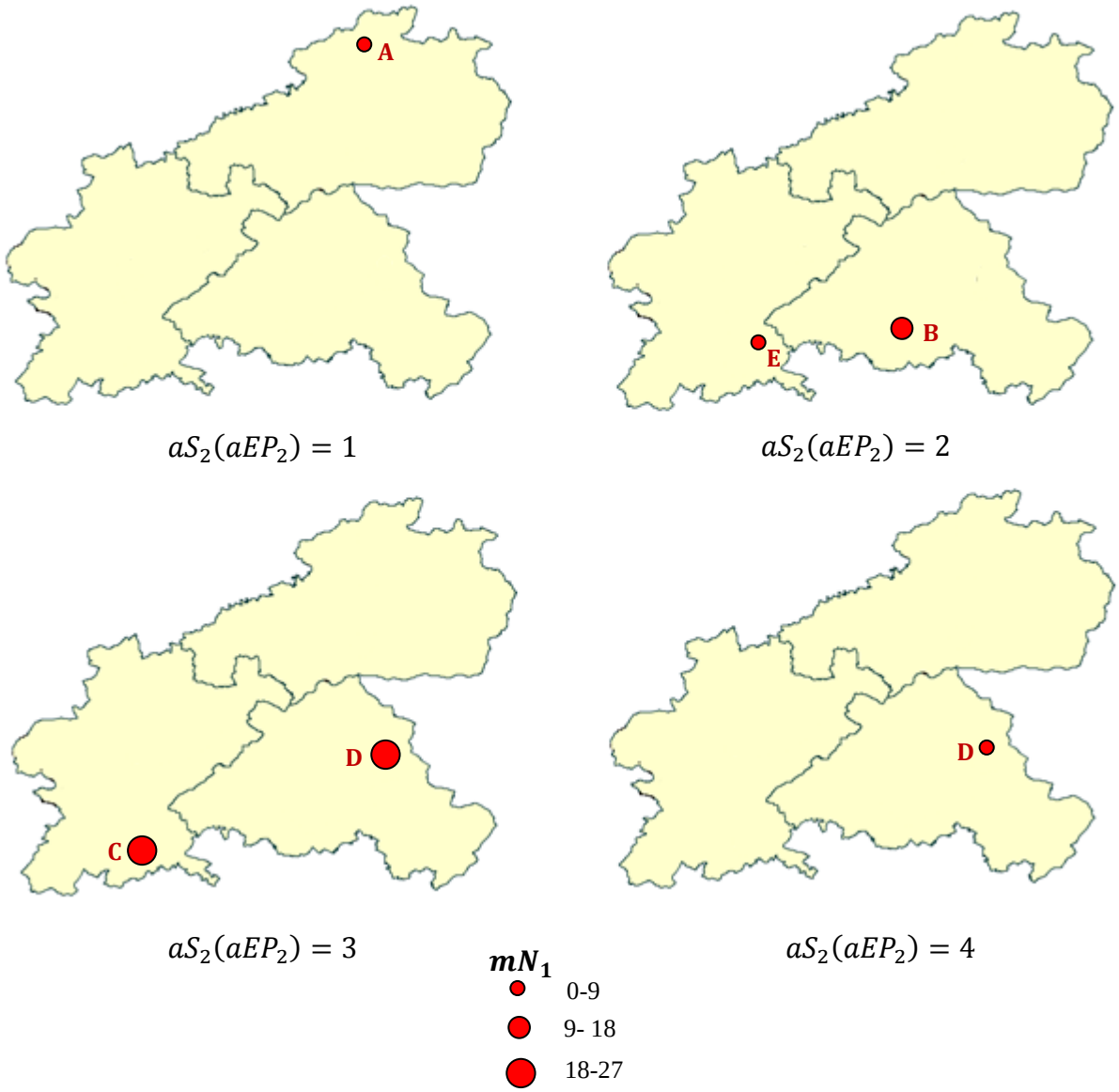


Figura 33 - Mapas que obter-se-ia ao realizar *slices* sobre $aS_2(aEP_2)$.

Através da Figura 33 fica mais explícito que sem a solução proposta torna-se difícil ter uma perspectiva global das relações de ambos os atributos espaciais.

Associado a este contexto de interacção surgem casos onde seria interessante analisar, por exemplo, "o total de passageiros que partiram do aeroporto *X* para o aeroporto *Y* num dado ano".

Neste contexto, uma operação de *drill-up* sobre qualquer um dos atributos espaciais poderá resultar numa situação em que um dos atributos é um ponto e o outro é um polígono, tal como se vai analisar na secção seguinte (3.2.2).

3.2.2 Ponto com Polígono

O segundo caso pode surgir devido a uma operação de *roll-up* proveniente da *situação 1* ou apenas quando se está a colocar os atributos espaciais (aEP_1 e aEP_2) em análise. Considere que os objectos do atributo aEP_1 são representados por pontos e os objectos do atributo aEP_2 correspondem a polígonos. O domínio de $\mathcal{D}(aS_1(aEP_1))$ mantém-se ao contrário do domínio de $\mathcal{D}(aS_2(aEP_2))$, que é definido por: $\mathcal{D}(aS_2(aEP_2)) = \{\text{Polígono}_X, \text{Polígono}_Y, \text{Polígono}_Z\}$.

Nesta situação, a função $mRF = f(aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2), mN_1)$ tem o seguinte resultado:

$aS_1(aEP_1)$	$aS_2(aEP_2)$	mN_1
A	Polígono_Y	4
B	Polígono_Z	10
C	Polígono_X	20
D	Polígono_X	26
E	Polígono_Z	2

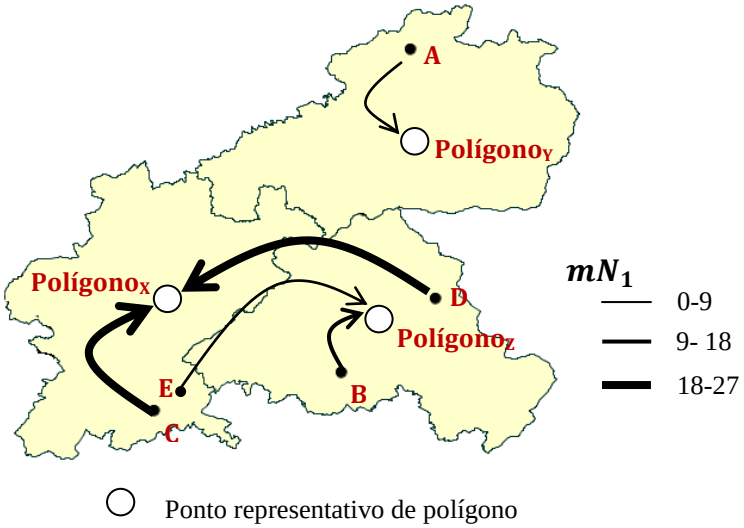


Figura 34 - Tabela de Suporte e respectivo resultado da função *mRF*.

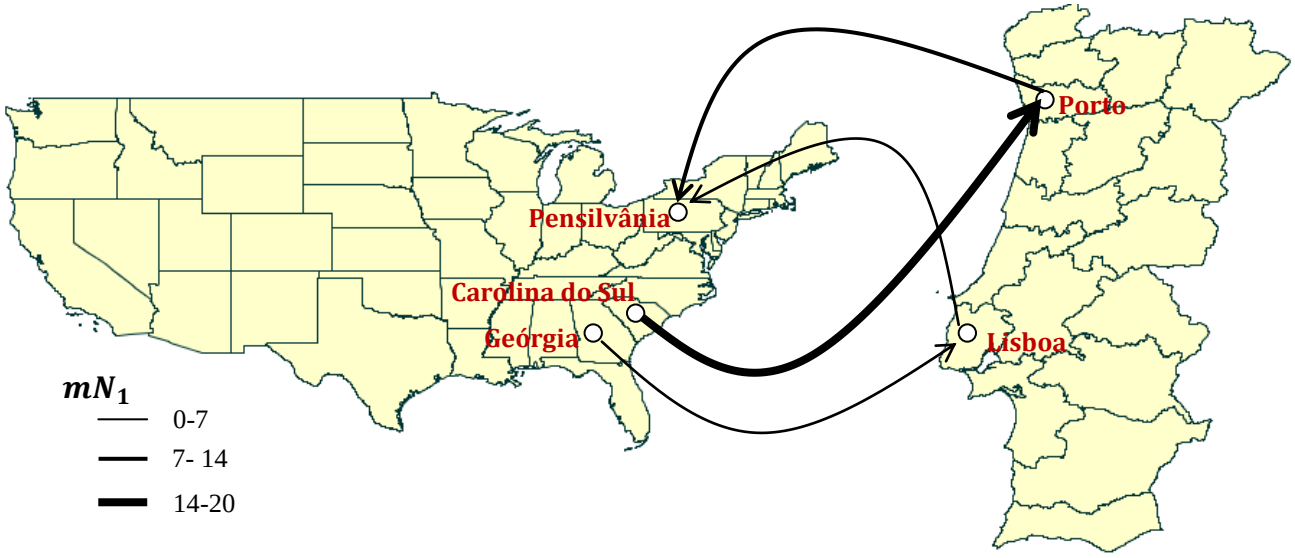
Anteriormente, os arcos que relacionavam os dois atributos espaciais eram os próprios objectos espaciais. No entanto, para lidar com este caso, é necessário criar um ponto representativo para cada polígono. Os pontos representativos que nos parecem mais adequados correspondem aos centróides destes. Assim, as extremidades dos arcos são os pontos do atributo aEP_1 e os centróides dos polígonos do atributo aEP_2.

Um exemplo concreto deste contexto de interacção é a análise do total de voos que tenham origem num aeroporto *X* para uma região ou país *P*.

3.2.3 Polígono com Polígono

É possível obter este caso tanto por via de uma operação de *drill-up* sobre o atributo aEP_1 dada a situação anterior (secção 3.2.2), como pela introdução dos atributos espaciais (aEP_1 e aEP_2) em análise. Todavia, não é suficiente realizar uma dedução directa e representar os centróides de cada polígono à semelhança do que é proposto no caso anterior (*Situação 2*). Em geral, deve-se representar os centróides, dos polígonos de cada atributo espacial, com diferentes representações.

Existem, no entanto, situações onde não existe essa necessidade. Nesses casos, a abordagem a seguir baseia-se apenas no uso de centróides como extremidades dos arcos, como pode ser visualizado na Figura 35. Considera-se $\mathcal{D}(aEP_1)$ os distritos de Portugal, enquanto que o $\mathcal{D}(aEP_2)$ corresponde aos estados dos EUA. Como os polígonos não partilham o mesmo espaço geográfico, não existe a necessidade de ter diferentes representações dos centróides. A função de representação $mRF = f(aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2), mN_1)$ tem o seguinte resultado:



$aS_1(aEP_1)$	$aS_2(aEP_2)$	mN_1
Porto	Pensilvânia	10
Lisboa	Pensilvânia	2
Carolina do Sul	Porto	18
Geórgia	Lisboa	6

Figura 35 - Tabela de Suporte e respectivo resultado da função *mRF*.

Agora considere o $\mathcal{D}(aEP_1) = \{\text{Polígono}_1, \text{Polígono}_2, \text{Polígono}_3, \text{Polígono}_4, \ldots, \text{Polígono}_n\}$ e o $\mathcal{D}(aEP_2)$ da secção 3.2.2, onde os polígonos de aEP_1 e de aEP_2 partilham o mesmo espaço de

contexto. Ao manter a aproximação anterior o resultado da função $mRF = f(aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2))$ seria o seguinte:

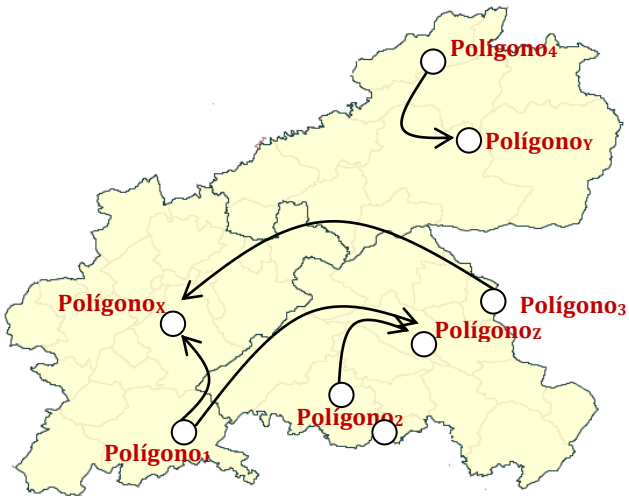


Figura 36 - Resultado da função *mRF* no segundo caso mantendo a aproximação inicial.

Como é observável através da Figura 36, a utilização da mesma representação visual dos centróides para os polígonos presentes no $\mathcal{D}(aEP_1)$ e no $\mathcal{D}(aEP_2)$ facilmente leva o utilizador a confundir a semântica associada aos centróides, isto é, não é facilmente visível se um centróide x representa um polígono do domínio de aEP_1 ou do domínio de aEP_2. Para tal não se verificar existe a necessidade de ter diferentes representações visuais dos centróides do domínio de $\mathcal{D}(aEP_1)$ e de $\mathcal{D}(aEP_2)$, como ilustra a Figura 37.

$aS_1(aEP_1)$	$aS_2(aEP_2)$	mN_1
$Polígono_4$	$Polígono_Y$	4
$Polígono_2$	$Polígono_Z$	10
$Polígono_1$	$Polígono_X$	20
$Polígono_3$	$Polígono_X$	26
$Polígono_1$	$Polígono_Z$	2

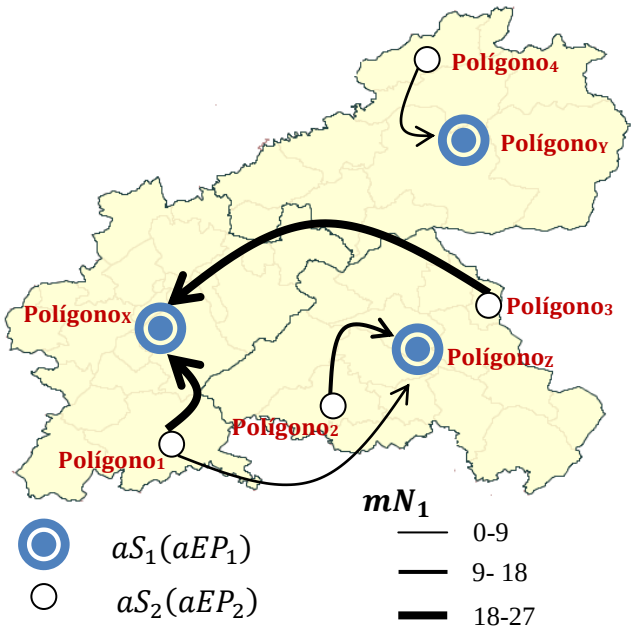


Figura 37 - Tabela de Suporte e respectivo resultado da função *mRF*.

Ao longo dos casos anteriores, a relação de 1:1 entre o mapa e a tabela de suporte é respeitada. Cada linha da tabela de suporte corresponde a um e só um arco do mapa. Contudo, estamos ainda a considerar contextos apenas com uma métrica. Assim, ao adicionar uma métrica numérica ao contexto anterior, a função $stRF = f(aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2), mN_1, mN_2)$ tem o seguinte resultado:

$aS_1(aEP_1)$	$aS_2(aEP_2)$	mN_1	mN_2
$Polígono_4$	$Polígono_Y$	4	21
$Polígono_2$	$Polígono_Z$	10	23
$Polígono_1$	$Polígono_X$	20	34
$Polígono_3$	$Polígono_X$	26	16
$Polígono_1$	$Polígono_Z$	2	1

Figura 38 - Tabela de suporte na presença de métricas numéricas.

De modo a manter a relação de 1:1 entre a tabela de suporte e o mapa, podem ser utilizadas diferentes formas de representação dos valores das métricas numéricas. Uma das formas de representação é a aplicação de duas variáveis visuais (luminosidade, espessura) mapeando cada métrica a uma dessas variáveis. Este assunto prende-se mais com questões de visualização e encontra-se abordado na secção Estilos e Legenda (secção 3.4) e em anexo (secção A.1).

Deste modo, com a utilização de duas variáveis visuais (luminosidade, espessura) o retorno da função $mRF = f(aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2), mN_1, mN_2)$ é o seguinte:

Figura 39 - Exemplo do resultado da função *mRF* com duas métricas numéricas.

Imagine que, após uma operação de *roll-up* sobre o primeiro atributo espacial aEP_1, nos deparávamos com o seguinte contexto: $\mathcal{D}(aS_1(aEP_1)) = \mathcal{D}(aS_2(aEP_2)) = \{\text{Polígono}_X, \text{Polígono}_Y, \text{Polígono}_Z\}$. Excluindo desta representação as métricas numéricas, a função $mRF = f\left(aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2)\right)$ retorna o seguinte resultado:

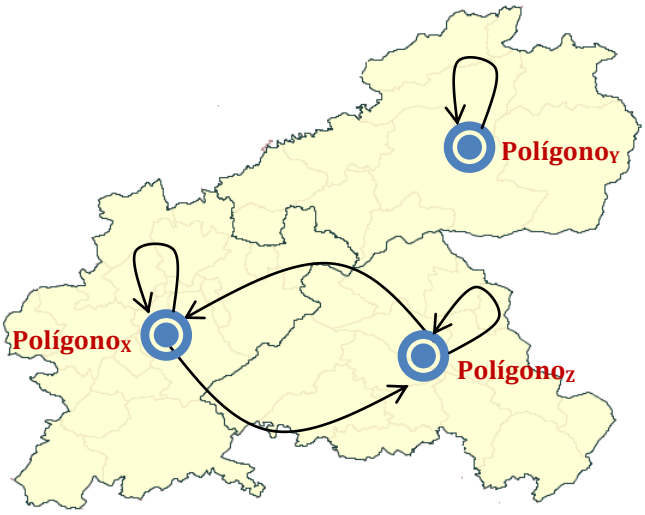


Figura 40 - Resultado da função *mRF* após uma operação de *roll-up*.

Como ilustra a figura anterior (Figura 40), os arcos cujas extremidades se encontram na mesma zona foram substituídos por um arco que aponta para a própria extremidade.

3.2.4 Ponto com Linha

Assume-se o conjunto de dados inicial, mas agora a representação do segundo atributo espacial (aEP_2) é a forma geométrica *linha*. A função $mRF = f(aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2))$ representa a relação entre os atributos espaciais (aEP_1 e aEP_2) através de uma linha orientada nos pontos onde se verifica a distância mínima entre os objectos espaciais (Figura 41).

$aS_1(aEP_1)$	$aS_2(aEP_2)$
A	1
B	2
C	3
D	4

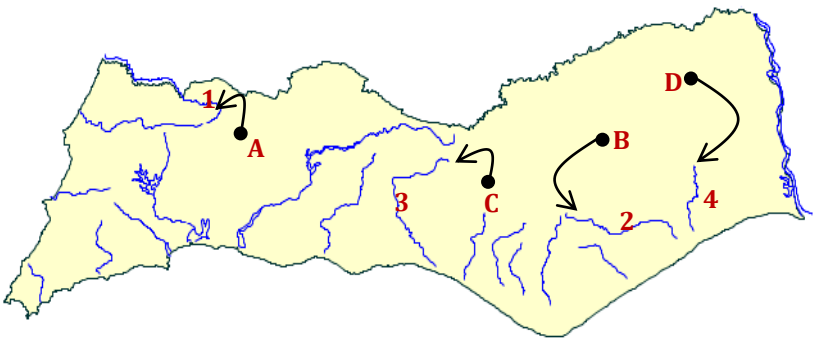


Figura 41 - Tabela de Suporte e respectivo Resultado da função *mRF*.

Novamente, a relação de 1:1 entre a tabela de suporte e o mapa é obtida, mantendo uma visão global das relações entre os dois atributos espaciais. Existem contextos onde este caso de interacção poderá mostrar-se muito útil, como por exemplo para o estudo da quantidade de resíduos lançados pelas indústrias nos rios.

Porventura, pode existir a necessidade de realizar uma operação de *roll-up* sobre qualquer atributo espacial (aEP_1 ou aEP_2). Tendo por base o exemplo anterior, uma operação de *roll-up* sobre o primeiro atributo espacial seria passar do nível da localização das indústrias para o nível de distritos. Já uma operação de *drill-up* sobre o segundo atributo espacial seria navegar do nível rios para o nível bacia hidrográfica. Em ambos os casos, o atributo espacial que se encontra no nível de granularidade acima da hierarquia espacial pode ser representado pelo tipo geométrico *polígono*.

Ao se constatar o facto anterior, ir-se-á cair num novo caso de interacção. Consoante o atributo ao qual é realizada a operação de *roll-up*, poder-se-á seguir para uma situação diferente. Considerando o exemplo anterior, no segundo caso seguia-se para a *situação 2*, e no primeiro caso ia-se para a *situação 5*, que será abordada na secção seguinte (secção 3.2.5).

3.2.5 Polígono com Linha

Esta situação verifica-se quando uma das formas geométricas dos atributos espaciais é a linha e a outra é o polígono. Neste caso, a função mRF irá combinar tanto características de representação na presença de polígonos como características da situação *Ponto com Linha* (secção 3.2.4), isto é, os pontos representativos dos polígonos são os centróides e os arcos entre os centróides e os objectos espaciais de aEP_2 são obtidos à distância mínima (Figura 42).

Considere o contexto anterior (secção 3.2.4), em que é realizada uma operação de *drill-up* sobre o atributo aS_1.

$aS_1(aEP_1)$	$aS_2(aEP_2)$
Polígono_X	1
Polígono_X	2
Polígono_X	3
Polígono_X	4

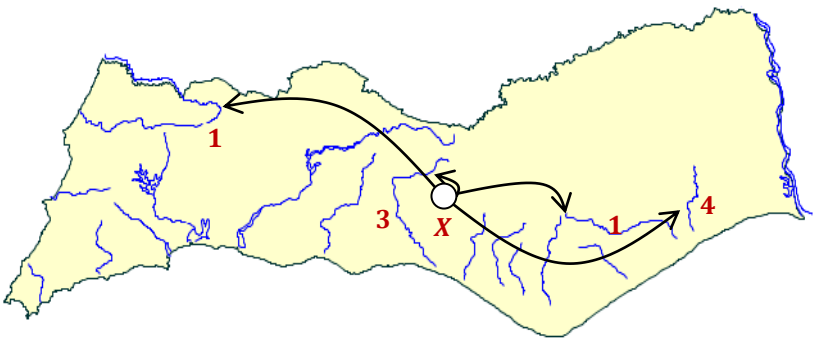


Figura 42 - Tabela de Suporte e respectivo Resultado da função *mRF*.

3.2.6 Linha com Linha

Por último, encontra-se a situação em que ambos os atributos espaciais são representados por linhas. De forma genérica, este caso de interacção será útil para realizar análises de fluxos, nomeadamente para detectar zonas de maior trânsito através da análise do número de viaturas que provêm de um troço para outro.

Nestes contextos de interacção, ambos os atributos espaciais estão geograficamente relacionados entre si. Com base no exemplo anterior, ambos os troços de estrada num determinado ponto têm que se intersectar.

Considere o seguinte contexto: $\mathcal{D}(aS_1(aEP_1)) = \mathcal{D}(aS_2(aEP_2)) = \{A, B, C, D, E\}$. A função $mRF = f(aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2))$ representa a relação entre os atributos espaciais (aEP_1 e aEP_2) através de uma linha que conecta ambos os objectos, mas ao contrário das representações anteriores, este arco encontra-se próximo da intersecção entre os atributos espaciais (Figura 43).

$aS_1(aEP_1)$	$aS_2(aEP_2)$
D	A
B	A
E	A
A	C

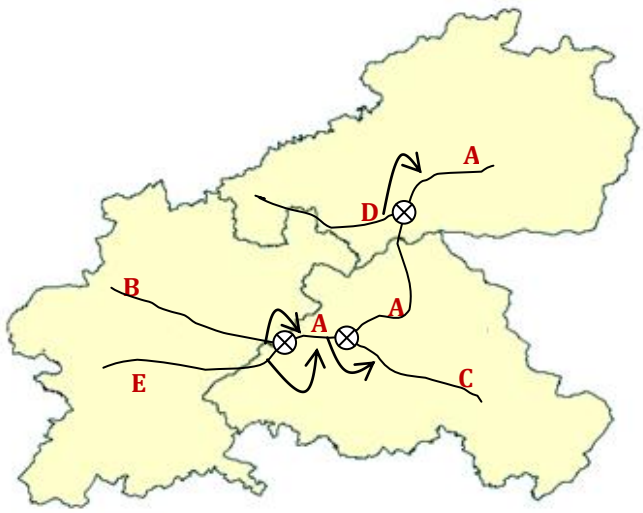


Figura 43 - Tabela de Suporte e respectivo Resultado da função *mRF*.

3.2.7 Interacção com o Utilizador

Relembro que um dos objectivos do modelo genérico SOLAP, que nos propomos a estender, é permitir análises exploratórias dos dados por parte dos utilizadores. Com este propósito, são apresentadas abaixo possíveis interacções que possam ocorrer durante uma análise com dois atributos espaciais de diferentes dimensões.

3.2.7.1 Interacção Utilizador - Mapa

Existem cenários onde pode surgir um número elevado de arcos, o que dificulta a visualização do mapa, tornando as análises pouco intuitivas. Para dar resposta a este problema, propomos duas abordagens: (*i*) permitir ao utilizador focar-se num determinado ponto; (*ii*) através de agrupamento espacial (ver secção 3.3).

Por focar num determinado ponto, entenda-se que o utilizador "selecciona" um determinado marcador x, presente no mapa, que consiste numa extremidade de um arco. Esta característica é ilustrada na Figura 44.

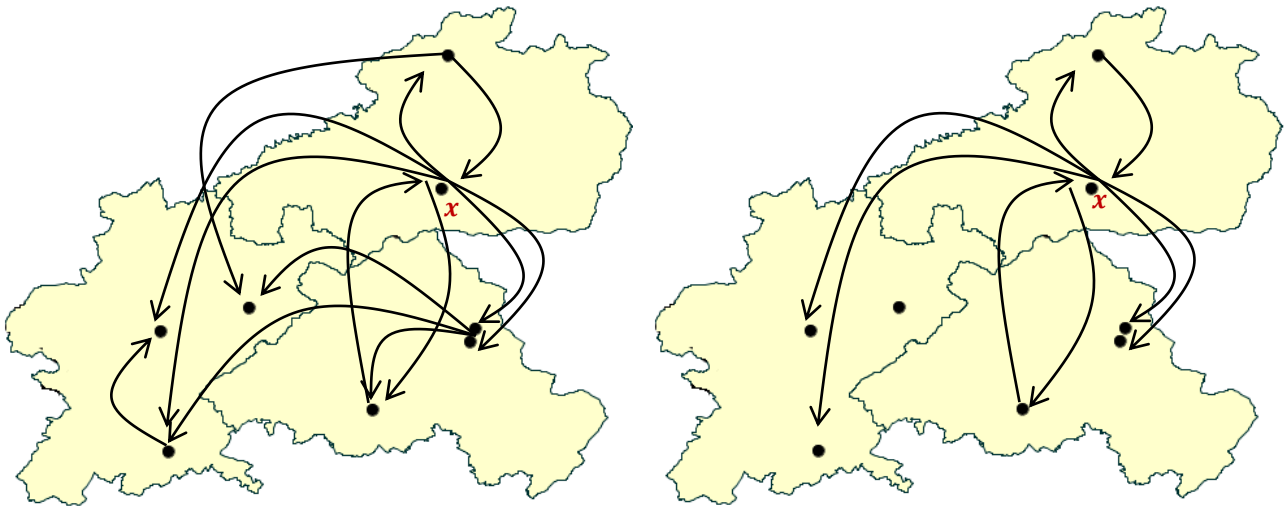


Figura 44 - Exemplo de selecção de uma determinada extremidade.

Após o utilizador se focar num determinado ponto x, todos os outros arcos serão "escondidos", apresentando apenas aqueles que estão relacionados com o ponto seleccionado. O utilizador pode ainda optar por apenas visualizar arcos com origem ou destino no ponto seleccionado. É importante referir que a informação disponibilizada na tabela de suporte se ajusta e apenas apresenta as linhas cujos dados estão a ser visualizados no mapa.

3.2.7.2 Interacção Utilizador - Gráfico de Suporte

Nesta secção é utilizado o contexto genérico da secção 3.2.3 (mapa da Figura 39 e tabela de suporte da Figura 38). Inicialmente, sugerimos que o gráfico de suporte por defeito disponibilize a informação completa da tabela de suporte, em formato de gráfico de barras, em que cada elemento do eixo das abcissas consiste na concatenação dos valores $aS_1(aEP_1)$ e $aS_2(aEP_2)$ presente numa linha da tabela de suporte, e os respectivos valores das métricas numéricas (mN_1, mN_2) estão

associados ao eixo da ordenada. O resultado da $scRF = f(aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2), mN_1, mN_2)$ é o seguinte:

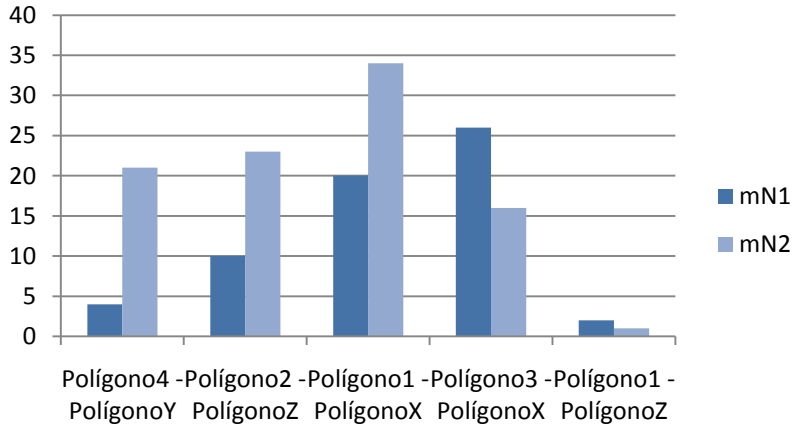


Figura 45 - Gráfico de Suporte por defeito.

A interacção prevista conjugando a tabela e o gráfico de suporte baseia-se na selecção de linhas, colunas ou um conjunto de células da tabela de suporte. Neste gráfico serão disponibilizados apenas os dados seleccionados. Além disso, existe a possibilidade de ordenação do gráfico por um dos atributos semânticos [$aS_1(aEP_1)$ ou $aS_2(aEP_2)$] ou por uma das métricas que esteja a ser visualizada (mN_1, mN_2).

3.2.7.3 Interacção Utilizador - Orientação da Linha

Ao longo das secções anteriores, tem sido indirectamente assumido que a orientação da linha, representada pela função mRF, era obtida da seguinte forma: o atributo semântico $aS_1(aEP_1)$ corresponde à origem do arco e o atributo semântico $aS_2(aEP_2)$ designa o destino do arco. Por vezes, a conexão dos diferentes objectos geográficos, ao ser realizada com uma linha orientada oferece uma forma intuitiva de realizar análises do género "o cliente *X* comprou o produto *Y* na loja *Z*". Todavia, a relação do atributo $aS_1(aEP_1)$ com $aS_2(aEP_2)$ nem sempre corresponde à semântica da análise que se está a realizar. Basta para isso considerar a análise "cliente *X* foi à loja *Y*" mas em que o atributo $aS_1(aEP_1)$ representa os nomes das lojas e $aS_2(aEP_2)$ representa os nomes dos clientes. Com o objectivo de adequar a análise ao processo cognitivo do utilizador, a definição das propriedades da orientação da linha que conecta ambos os atributos espaciais deve ser definida pelo utilizador. Assim, associado a este caso de interacção, no modelo genérico deverá existir uma forma para definir:

- Se a linha é orientada ou simples;
- No caso de ser orientada, permitir definir a origem e o destino.

Repare que esta interactividade só deve ser permitida quando existe uma relação disjunta dos domínios dos atributos $aS_1(aEP_1)$ e $aS_2(aEP_2)$. Quando não se verifica esta relação, o arco deve ser obrigatoriamente orientado. Se assim não fosse, ao analisar, por exemplo, o número de voos entre Portugal e o EUA, visualmente não seria distinguível o arco referente à relação Portugal – EUA do arco que se refere a relação EUA – Portugal. No entanto, o utilizador poderá estar apenas interessado na relação Portugal – EUA (e não no sentido das trocas), e, quando assim é, deve ser dada a possibilidade ao utilizador para que as métricas associadas à relação Portugal - EUA e EUA - Portugal sejam agregadas numa só linha.

3.2.8 Atributos Semânticos de Dimensões Semânticas

Aos casos anteriores podem ainda ser adicionados atributos semânticos de dimensões semânticas e/ou de dimensões espaciais. Apesar de já existirem propostas de Ruben Jorge [**1**] para lidar com estes atributos semânticos na presença de um atributo espacial, é necessário rever estes casos na presença de dois atributos espaciais de diferentes dimensões e analisar de que forma afecta a relação de 1:1 entre a tabela de suporte e o mapa.

Neste caso é definido como vão ser apresentados os dados quando se introduzem atributos semânticos de dimensões semânticas a análises que apenas incluem métricas numéricas e os próprios atributos espaciais. Ao vector de objectos representativo inicial (secção 3.2) são adicionados os atributos semânticos aS_i . Desta forma, o vector de objectos representativo apresenta a seguinte estrutura:

$$vObj = \{aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2), mN_1, \dots, mN_n\ aS_1, \dots, aS_i\}.$$

Quando se está na presença de atributos semânticos de dimensões semânticas, para cada combinação dos valores dos atributos $aS_1(aEP_1)$ e $aS_2(aEP_2)$ haverá diversos valores de aS_i . Em consequência, a relação de 1:1 entre a tabela de suporte e o mapa não é respeitada. Para solucionar este problema, mas com apenas um atributo espacial, Ruben Jorge [**1**] introduziu as tabelas pivô com restrições (referidas anteriormente, na secção 3.1). Com dois atributos espaciais mantém-se a estrutura da tabela de suporte.

Seja o atributo semântico (aS_x) introduzido na tabela de suporte (Figura 38) com o seguinte domínio: $\mathcal{D}(\mathrm{aS_X}) = \{\mathrm{valor}_1, \mathrm{valor}_2\}$.

O resultado parcial da função $stRF = f(aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2), mN_1, mN_2, sA_x)$ é o seguinte (Figura 46):

		$\mathbf{valor_1}$		$\mathbf{valor_2}$	
$\mathbf{aS_1(aEP_1)}$	$\mathbf{aS_2(aEP_2)}$	$\boldsymbol{mN_1}$	$\boldsymbol{mN_2}$	$\boldsymbol{mN_1}$	$\boldsymbol{mN_2}$
Polígono_4	Polígono_Y	8	23	23	32

Figura 46 - Tabela de suporte parcial resultante da função *stRF* com aS_X.

Com o objectivo de disponibilizar a informação no mapa presente na tabela de suporte, respeitando a propriedade 1:1, propomos a utilização de gráficos.

Para efeitos de ilustração vamos assumir que o atributo aS_X representa um atributo temporal (ex: valor_1 = 2008 e valor_2 = 2009). Neste caso, utilizaremos o gráfico de linha empilhados com marcadores, em que o eixo das abcissas contém os valores do $\mathcal{D}(aS_X)$ em análise e o eixo das ordenadas corresponde aos valores das métricas. Cada linha está associada a uma métrica numérica. Deste modo, $mRF = f(aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2), mN_1, mN_2, aS_x)$ retorna:

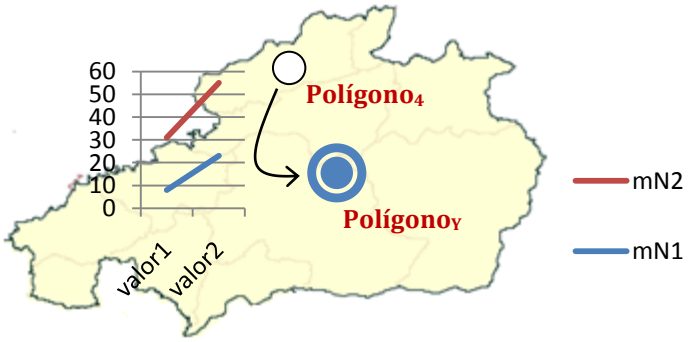


Figura 47 - Resultado da função *mRF* com atributos semânticos de dimensões semânticas.

Como é possível observar através da Figura 47, é mantida a relação 1:1 entre a tabela de suporte e o mapa. Por outro lado, é possível ter uma visão global, quer da evolução das métricas ao longo dos valores semânticos, quer das possíveis correlações que possam existir entre as métricas. Porém, um número elevado de arcos implicará um número elevado de gráficos, o que torna o mapa desorganizado. Novamente, para resolver essa questão propomos as seguintes formas: (*i*) só é disponibilizado o gráfico se o utilizador se focar no determinado arco; (*ii*) através de técnica de agrupamento espacial (ver secção 3.3).

Este caso de interacção poderá mostrar-se muito útil, por exemplo, para examinar a evolução das exportações/importações entre países ao longo dos anos.

Além do gráfico de linha empilhado com marcadores, outros gráficos podem ser solução para a representação dos valores das métricas. Por outro lado, tem que se ter em conta que podemos ter mais do que um atributo semântico em análise. Estes tópicos estão mais relacionados com a visualização e são discutidos na secção Estilos e Legenda (secção 3.4) e em anexo (secção 3.41).

3.2.9 Atributos Semânticos de Dimensões Espaciais

No caso anterior (secção 3.2.8) é definido de que forma são apresentados os dados na presença de atributos semânticos de dimensões semânticas, mas, neste caso, estes provêm de dimensões espaciais. Os vectores de objectos associados a este caso de interacção mantêm a estrutura, como pode ser observado pelo vector representativo:

$$vObj = \{aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2), mN_1, \dots, mN_n, aS_1, \dots, aS_i\}$$

Quando se adiciona um atributo semântico de uma dimensão espacial, é necessário considerar o nível de granularidade do atributo aS_i relativamente ao atributo $aS_i(aEP_i)$. Caso o atributo aS_i se encontre no mesmo nível ou num nível superior de granularidade, então existe um e só um valor do atributo aS_i para cada $aS_i(aEP_i)$. Caso contrário, existem potencialmente vários valores.

Por outro lado, é necessário ter em conta que poderão existir atributos semânticos de cada dimensão espacial ou de ambas as dimensões espaciais. De modo a analisar os casos mais relevantes, vamos assumir que apenas é possível analisar um atributo semântico de cada dimensão espacial. Mais, que um atributo semântico de cada dimensão espacial torna-se mais um problema de visualização/representação dos dados da tabela de suporte. Esta questão é discutida na secção Estilos e Legenda (secção 3.4) e em anexo (secção 3.41).

Posto isto, podem surgir os seguintes casos (Tabela 3) em que a coluna *condição* denota a relação de granularidade entre os atributos:

	aS_x	Condição	$aS_1(aEP_1)$	aS_y	Condição	$aS_2(aEP_2)$
1	Em análise	≥	Em análise	Inexistente	-	Em análise
2	Em análise	≥	Em análise	Em análise	≥	Em análise
3	Em análise	≥	Em análise	Em análise	≤ ou incomparável	Em análise
4	Em análise	≤ ou incomparável	Em análise	Inexistente	-	Em análise
5	Em análise	≤ ou incomparável	Em análise	Em análise	≤ ou incomparável	Em análise

Tabela 3 - Tabela com os diferentes casos possíveis neste contexto.

Nas secções subsequentes serão abordados os diferentes casos em mais detalhe. Considere os dados genéricos apresentados na secção 3.2.

3.2.9.1 Atributos Semânticos ao mesmo ou maior nível do que os Atributos Espaciais

O *caso 1* é muito semelhante ao *caso 2*, com a diferença de apenas existir um atributo semântico de uma dimensão espacial em análise. Deste modo, vamos abordar o *caso 2*.

Após adicionar ambos os atributos semânticos (aS_x e aS_y) , o resultado da função de representação da tabela de suporte ($stRF = f(aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2), aS_x, aS_y)$ é o seguinte:

$aS_1(aEP_1)$	aS_x	$aS_2(aEP_2)$	aS_y
A	X	1	P
B	Y	2	Q
C	Z	3	R
D	X	4	Q

Figura 48 - Tabela de suporte após a inserção do atributo semântico (aS_1).

O facto de ambos os atributos estarem a um nível superior ou igual comparativamente aos seus $aS_i(aEP_i)$ permite manter o número de linhas sem a utilização da tabela em forma de pivô.

Para realizar a representação visual destes atributos, basta atribuir a cada um uma propriedade visual (ex: cor, forma). Assim, $mRF = f(aEP_1, aS_1(aEP_1), aEP_2, aS_2(aEP_2), aS_x, aS_y)$ pode retornar o seguinte resultado:

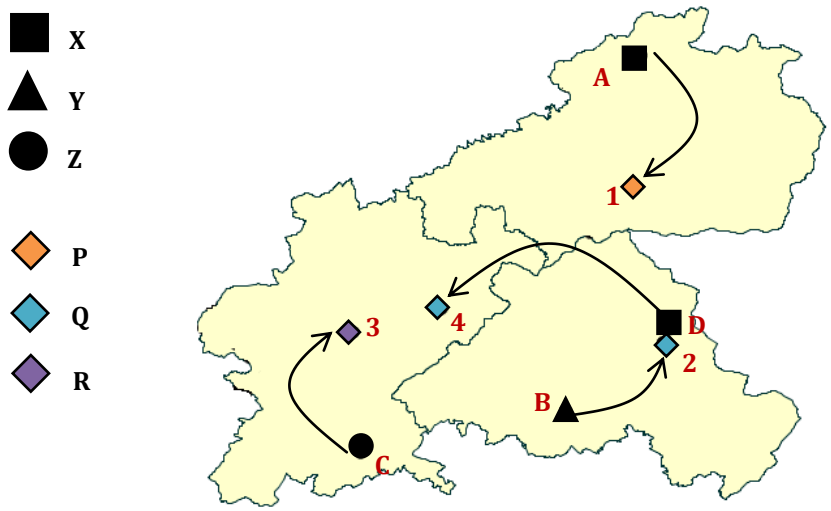


Figura 49 - Mapa Resultante da função de representação *mRF*.

Existem outras formas para realizar a representação visual dos atributos semânticos. Mais uma vez, este assunto prende-se mais com questões de visualização, que são discutidas na secção Estilos e Legenda (secção 3.4) e em anexo (secção 3.41).

3.2.9.2 Atributos Semânticos a um nível menor ou incomparável do que os Atributos Espaciais

Seja o atributo aS_1 o atributo que se encontra ao mesmo ou num nível superior de granularidade e o atributo aS_2 o atributo que se encontra a um nível de granularidade abaixo ou incomparável. O resultado da função de representação mRF, para o *caso 3*, será uma combinação da representação do caso anterior (secção 3.2.9.1) com a representação prevista no caso de interacção 3.2.8 (atributos semânticos de dimensões semânticas).

Por fim, quando ambos os atributos semânticos (aS_1 e aS_2) se encontram a um nível incomparável ou abaixo dos atributos espaciais (aEP_1 e aEP_2), significa que existem vários valores de aS_1 e aS_2 para cada combinação de $aS_1(aEP_1)$ e $aS_2(aEP_2)$. Deste modo, este caso (*Caso 5*) comporta-se da mesma forma que em 3.2.8. O mesmo se verifica para o *caso 4*.

3.3 Integração de Agrupamento Espacial

A integração do agrupamento espacial passa, inicialmente, por introduzir um modelo de agrupamento espacial dentro da fase de pré-processamento. Assim, de modo a oferecer uma solução de agrupamento espacial, para os objectos espaciais pontos e polígonos, foi definido um módulo de agrupamento espacial. Esse corresponde ao apresentado na figura abaixo (Figura 50).

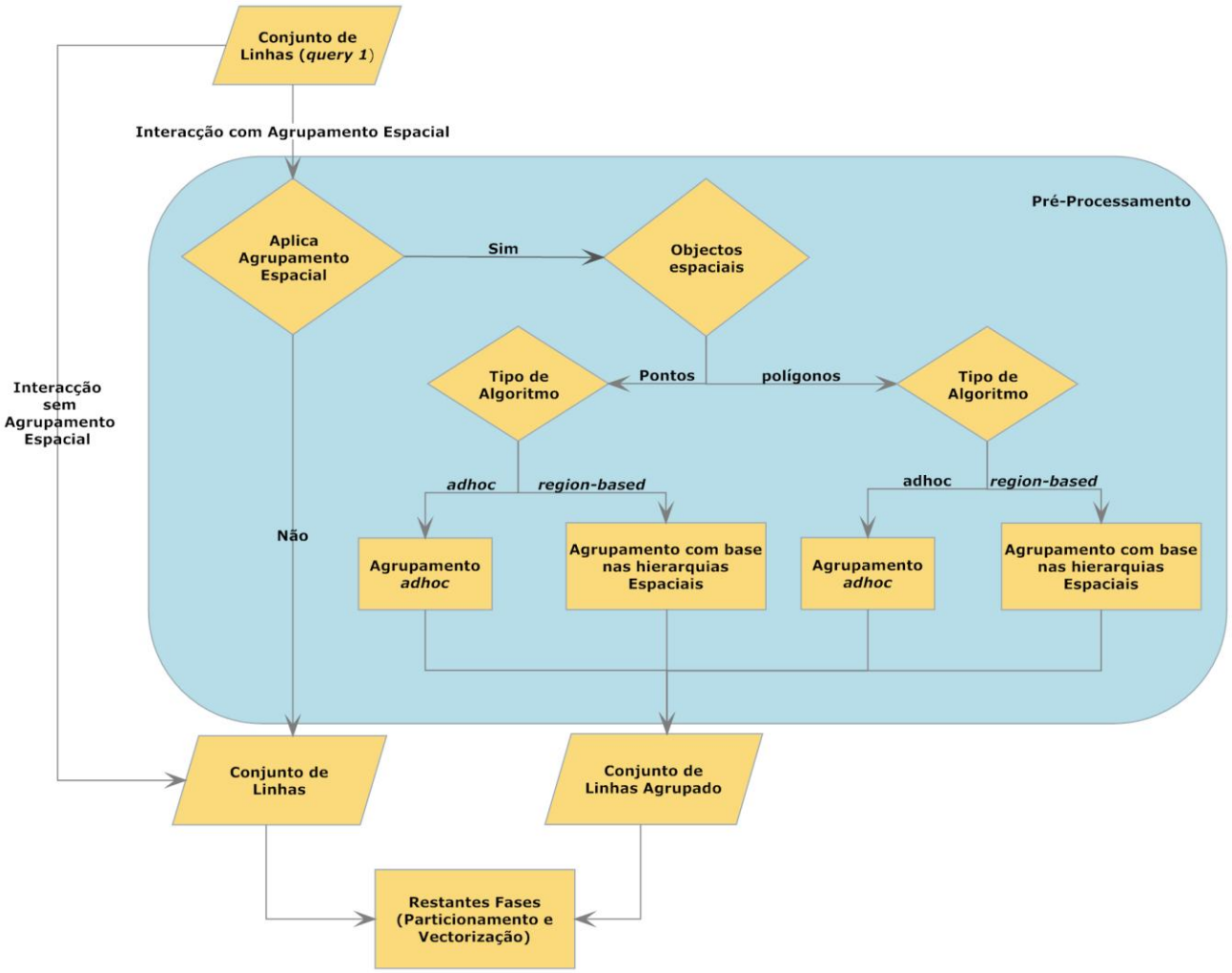


Figura 50 - Modelo de Pré-Processamento.

O módulo de pré-processamento tem como responsabilidade identificar o tipo de objectos espaciais associados à interrogação realizada. Consoante o tipo de objectos, aplica um algoritmo de agrupamento espacial adequado com base nos atributos espaciais. O utilizador pode ainda definir qual o tipo de algoritmo que se pretende (*adhoc* ou o algoritmo com base em hierarquias espaciais).

No entanto, antes de se realizar todo este fluxo referido anteriormente, é importante avaliar se este é realmente necessário. Assim, numa fase inicial é importante avaliar se existem argumentos suficientes para garantir que a computação adicional que se introduz é benéfica, ou seja, se se justifica a aplicação de um algoritmo de agrupamento. Nas secções subsequentes, serão apresentadas as diferentes fases deste modelo de pré-processamento em mais pormenor.

3.3.1 Avaliar Conjunto de Linhas

O objectivo nesta fase é decidir se a introdução do processamento adicional que é requerido pelos algoritmos de agrupamento espacial vai trazer algum benefício.

Da perspectiva do utilizador, com a introdução de agrupamento espacial sobre o conjunto de dados inicial podem-se verificar dois benefícios, quer separadamente, quer em simultâneo: melhoria do desempenho e boa visibilidade dos objectos presentes no mapa.

Assim, devem ser determinados indicadores sob estes dois pontos de vistas. Estes indicadores são igualmente importantes e, portanto, basta um destes indicadores ser positivo para que seja realizada a fase de pré-processamento.

3.3.2 Agrupamento de Pontos

O agrupamento de pontos pode ser realizado sob duas formas: *adhoc* ou com base nas hierarquias espaciais. O agrupamento *adhoc* cria grupos de pontos onde os quais não têm qualquer significado semântico para além da proximidade geográfica. Ao contrário do agrupamento *adhoc*, o agrupamento com base nas hierarquias espaciais, apesar de criar grupos com base na proximidade geográfica, restringe-os pelo nível da hierarquia. Nas secções abaixo será detalhado o modelo para cada uma das formas de agrupamento.

3.3.2.1 Agrupamento *adhoc*

Após obter o conjunto de linhas inicial, subsistem dois espaços sobre os quais o agrupamento espacial se pode basear: utilizar coordenadas latitude ou longitude (mundo real) ou recorrer às coordenadas da projecção geográfica do mundo real (*canvas*).

Um algoritmo de agrupamento espacial *X* deverá realizar o agrupamento através das coordenadas da projecção geográfica do mundo real. Primeiro, a alternativa não considera a área do *canvas*. Este facto não deve ser ignorado, pois ao fixar um determinado conjunto de dados, quanto maior a área de *canvas* menor será a densidade dos pontos (entenda-se marcadores). Segundo, o nível de zoom deve também ser considerado, o que não é verificado com as coordenadas do mundo real.

Considere o seguinte conjunto de linhas associado à sua representação dos objectos no mapa (mRF):

aEP	$aS(aEP)$	mN_1
$Ponto_1$	A	12
$Ponto_2$	B	32
$Ponto_3$	C	74
$Ponto_4$	D	34
$Ponto_5$	E	15
$Ponto_6$	F	57

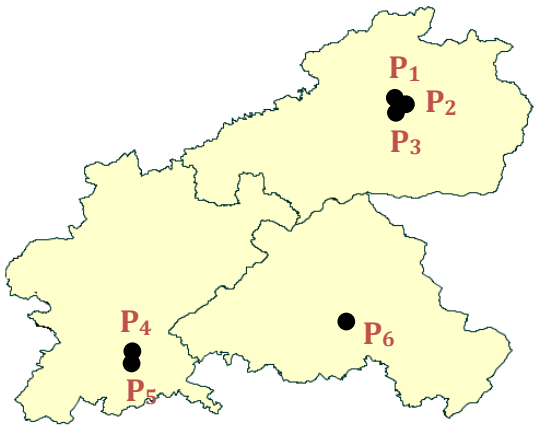


Figura 51 - Conjunto de linhas inicial e respectivo mapa.

Inicialmente são obtidas as coordenadas (x, y) de todos os objectos geográficos presentes no conjunto de linhas. O conjunto de coordenadas é designado de vector de pontos (vP).

De seguida, é aplicado um algoritmo X (de agrupamento espacial) ao vector de pontos ($X(vP)$). O resultado do algoritmo será os pontos associados a um ou a nenhum grupo. Considere que, para o algoritmo X aplicado ao conjunto de linhas inicial, o resultado é o seguinte: $g_1 = \{P_1, P_2, P_3\}$, $g_2 = \{P_4, P_5\}, noise = \{P_6\}$. Ao conjunto de grupos chamamos *vector de grupos* (vG).

Após a identificação dos grupos de pontos, é necessário realizar duas tarefas:

- Definir o ponto representativo de cada grupo;
- Agregar os dados pertencentes a um grupo.

Deste modo, o fluxo do modelo para o agrupamento de pontos *adhoc* é o seguinte:

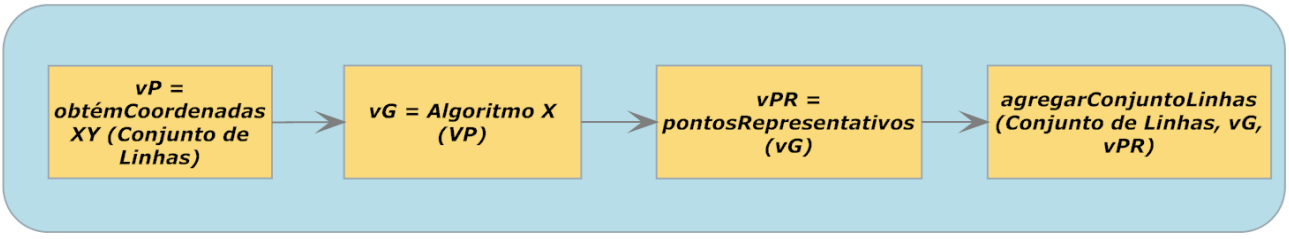


Figura 52 - Fases internas para o agrupamento de pontos *adhoc*.

Assim, após o exemplo anterior estar sujeito à fase de pré-processamento, o conjunto de linhas e respectivo mapa resultante está ilustrado na figura seguinte (Figura 53). Neste exemplo, foi utilizado o operador de agregação soma.

aEP	$aS(aEP)$	mN_1
Centróide$_1$	G1	118
Centróide$_2$	G2	49
Ponto$_6$	F	57

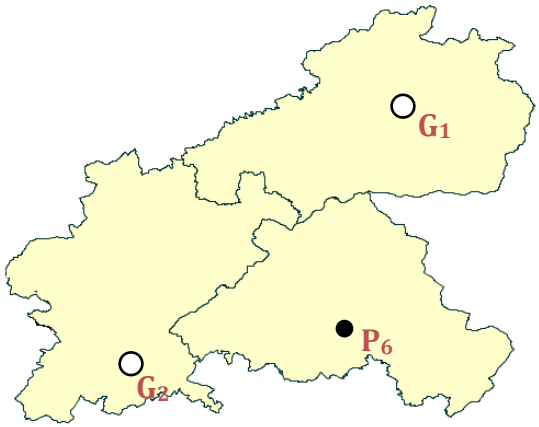


Figura 53 - Resultado do processo de agrupamento *adhoc* de pontos.

O exemplo, ainda que genérico, reflecte apenas o cenário elementar. É necessário considerar os restantes casos de interacção previstos no modelo genérico SOLAP e verificar as possíveis incompatibilidades que possam existir (ver secção 3.3.2.3).

3.3.2.2 Agrupamento com base nas Hierarquias Espaciais

O agrupamento dinâmico com base nas hierarquias espaciais consiste num agrupamento que combina a informação do nível de zoom com uma hierarquia espacial, pertencente à dimensão espacial em análise.

De forma análoga ao que era realizado com o agrupamento *adhoc*, é necessário obter as coordenadas x e y para cada objecto espacial (ponto) presente no conjunto de linhas. A fase seguinte consiste em aplicar um algoritmo de agrupamento espacial ao vector de pontos (vP) com o objectivo de encontrar as zonas de grande densidade, obtendo-se o vector de grupos (vG). Porém, para restringir os grupos pelas hierarquias espaciais, é necessário introduzir uma nova fase.

Seja dEP a dimensão espacial presente no conjunto de linhas. dEP contém $h_1, \dots, h_n$ hierarquias espaciais. Considere uma hierarquia h_x escolhida pelo utilizador, em que $i \leq x \leq n$. A hierarquia h_x é composta por $1, \dots, w$ níveis de granularidade, cujos níveis vão desde uma granularidade mais fina (nível 1) para um nível mais graúdo (nível w). Dos w níveis presentes na hierarquia h_x, só um nível a partir de i, representado por polígonos, pode ser utilizado para restringir os grupos de modo a que se obtenha um agrupamento baseado em regiões.

Para combinar tanto o nível de zoom do mapa como a hierarquia espacial h_x, os diversos níveis de zoom definidos para a zona do mapa são divididos igualmente pelos níveis contidos na hierarquia h_x, considerando apenas os níveis de $i, \ldots, w$ (Figura 54).

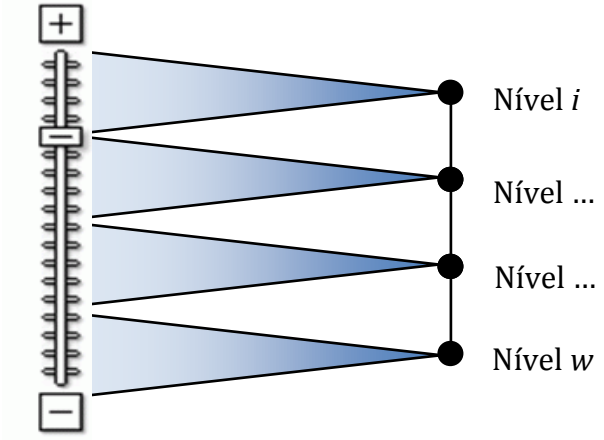


Figura 54 - Ilustração do mapeamento entre os níveis da hierarquia e os níveis de zoom.

Depois de se obter o vector de grupos (vG), é utilizada a informação definida anteriormente, e é extraído de cada grupo os subgrupos que partilhem o mesmo objecto espacial do nível pelo qual se está a restringir os grupos. A função responsável por realizar esta computação é a seguinte: $restringirGrupos(vG, h_x, nívelZoom)$.

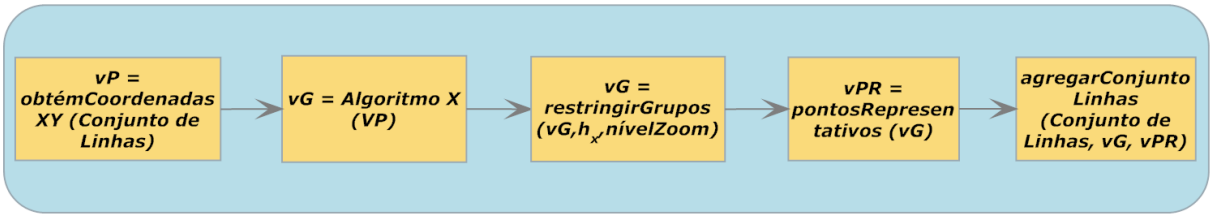


Figura 55 - Fases internas para o agrupamento base nas hierarquias espaciais.

Assim, o agrupamento dinâmico com base nas hierarquias espaciais é definido por um modelo semelhante ao modelo de agrupamento *adhoc*, mas com a introdução de uma nova fase, como pode ser observado na Figura 55.

3.3.2.3 Agrupamento de Pontos *versus* Casos de Interacção

Independentemente do tipo de agrupamento realizado a um determinado conjunto de linhas, nem sempre são lineares alguns aspectos envolventes neste processo.

Considere o *caso 4*, em que está presente pelo menos um atributo semântico de uma dimensão espacial, cuja granularidade está ao mesmo nível do atributo espacial já presente em análise.

Neste caso, realizar o agrupamento de pontos sem qualquer restrição pode potencialmente provocar a perda de informação. Ao adicionar um atributo semântico ao contexto inicial em análise colocado na secção 3.3.2.1 (agrupamento *adhoc*), obtemos o seguinte conjunto de linhas e respectivo mapa:

aEP	$aS(aEP)$	aS_1	mN_1
P_1	A	X	12
P_2	B	Y	32
P_3	C	Y	74
P_4	D	Z	34
P_5	E	X	15
P_6	F	Z	57

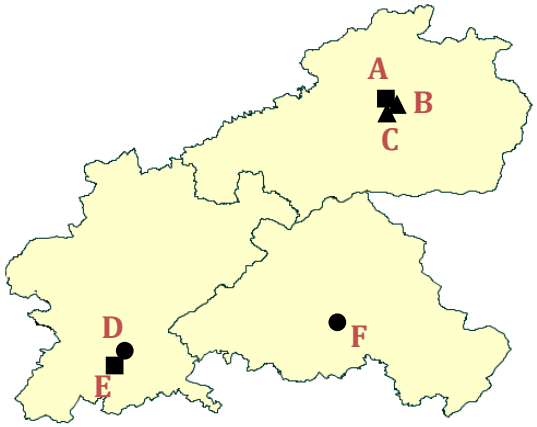


Figura 56 - Conjunto de linhas inicial e respectivo mapa resultante.

Realizar o agrupamento sem qualquer restrição provoca perca de informação relativamente ao atributo semântico aS_1. Ao agrupar os pontos P_1, P_2, P_3 e os pontos P_4, P_5 leva a que, visualmente, o utilizador não consiga identificar qual o valor do atributo semântico envolvente em cada grupo.

Quando é aplicado o agrupamento espacial nos diversos casos de interacção, não só a propriedade de 1:1 entre a tabela de suporte e o mapa deve ser salvaguardada, como também deverá existir a preocupação de manter a análise com as mesmas propriedades que se verificavam quando não ocorria agrupamento. Com este facto em mente, no *caso 4* de interacção (quando existem atributos semânticos a nível de granularidade superior):

1. Só é agrupado o ponto p com o ponto q caso estes partilhem os mesmos valores dos respectivos atributos semânticos (aS_i);
2. Por escolha do utilizador:
    a. Pode automaticamente a coluna do atributo aS_i ser excluída da análise;
    b. A coluna do atributo aS_i é colocada como cabeçalho.

Para o conjunto de dados anterior o resultado do agrupamento *adhoc* (verificando a primeira condição) é o seguinte:

aEP	$aS(aEP)$	aS_1	mN_1
$Ponto_1$	A	X	12
$Centróide_1$	G_1	Y	106
$Ponto_4$	D	Z	34
$Ponto_5$	E	X	15
$Ponto_6$	F	Z	57

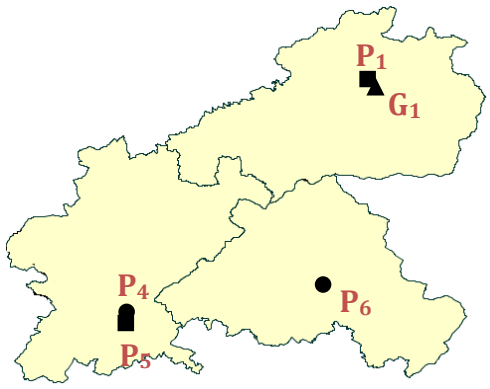


Figura 57 - Conjunto de linhas inicial e respectivo mapa resultante.

Nos outros casos não existem quaisquer restrições na presença de agrupamento de pontos. De notar que, mesmo nos casos de interacção, onde estão presentes dois atributos espaciais de diferentes dimensões em análise, o agrupamento é também um agrupamento de pontos. No entanto, só podem ser agrupados pontos do mesmo atributo espacial aEP_i.

3.3.3 Agrupamento de Polígonos

O processo de agrupamento de polígonos é semelhante ao processo de agrupamento *adhoc* dos pontos. O modelo tem o seguinte fluxo:

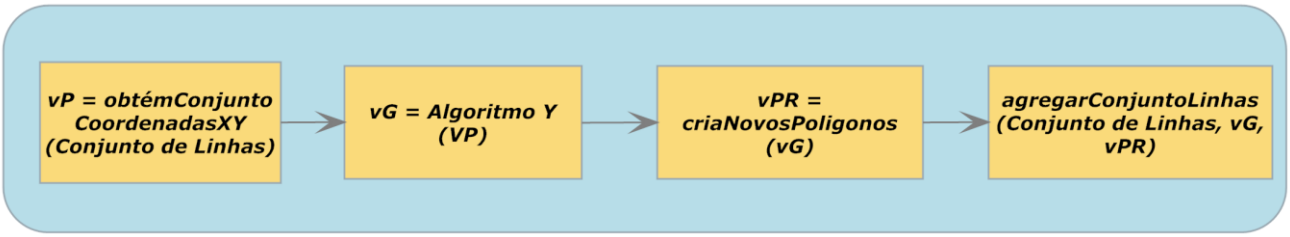


Figura 58 - Modelo representativo do processo de agrupamento de polígonos.

Inicialmente, o vector de polígonos (vP) é obtido de forma semelhante no modelo de agrupamento *adhoc*. Posteriormente, é aplicado um algoritmo de agrupamento espacial adequado para polígonos. Após obter o vector de grupos (vG) da execução do algoritmo Y, é realizado o processo de transformação do conjunto de linhas à semelhança do que era efectuado nos outros tipos de agrupamento definidos anteriormente.

O agrupamento de pontos fazia total sentido, pois os marcadores ao se sobreporem uns com os outros dificultavam a sua visualização e esse facto tinha repercussões negativas na visualização. Mas, ao contrário dos "pontos", os polígonos são objectos geográficos que não se sobrepõem.

Num contexto SOLAP, é típico a utilização de gráficos (barras, circulares, etc.) sobrepostos no mapa. Assim, considere o *caso 2* de interacção (uma dimensão espacial e múltiplas métricas):

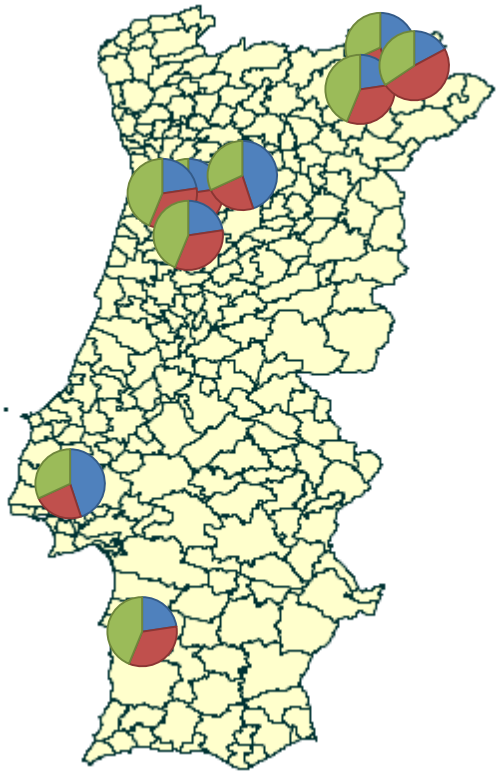


Figura 59 - Resultado da função mRF para um caso 2 de interacção.

Como é visível, existe desorganização no mapa. Não é fácil, ou mesmo impossível, a leitura de alguns gráficos circulares. Deste modo, pode-se concluir que a desorganização não é provocada directamente pelos polígonos, mas possivelmente pelos gráficos associados a estes.

Assim, o agrupamento *adhoc* de polígonos será benéfico em casos de interacção com gráficos associados a polígonos que possam trazer complicações no processo de análise. Isto é particularmente verdade nos *casos 2* e *3* quando o atributo espacial tem a forma geométrica polígono.

O agrupamento de polígonos com base nas hierarquias espaciais segue os mesmos princípios do agrupamento de pontos com base nas hierarquias espaciais.

3.3.4 Formas de Representação de Grupos

Até a este momento, não se tem discutido a representação dos grupos. No caso do agrupamento de pontos tem-se ilustrado a representação dos grupos através do centróide dos pontos de cada grupo e no caso do agrupamento de polígonos tem-se assumido a união dos polígonos de cada grupo.

Um possível exemplo da utilização do agrupamento espacial é a identificação dos locais onde existe maior actividade criminal. Assim, seria interessante identificar quais as áreas com mais incidentes criminais, de forma a poder definir uma estratégia adequada por parte da polícia.

Se neste contexto fosse utilizado o centróide como representação dos grupos resultantes de uma técnica de agrupamento espacial, dificilmente se podia obter uma visão real das áreas com maiores níveis de criminalidade. Uma representação interessante seria sim representar o grupo pelo menor polígono que contivesse todos os pontos pertencentes ao grupo (como ilustrado na Figura 60), onde, neste caso, cada ponto representa um incidente criminal.

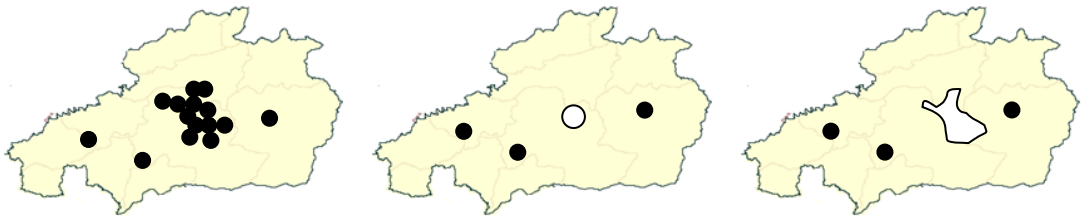


Figura 60 - Ilustra diferentes representações para um grupo de pontos.

Agora, considere que cada ponto denota uma indústria e o utilizador está interessado em analisar a quantidade de emissão. Numa situação destas, ao ter diferentes formas geométricas para representar os grupos, podem-se ter incompatibilidades nos estilos que estão a ser utilizados para mapear os valores das métricas. Por exemplo, se utilizarmos o tamanho para representar visualmente os valores da quantidade de emissão, torna impossível a utilização do mesmo estilo para as duas formas geométricas.

A representação que é utilizada para representar os grupos deve depender do contexto e da análise que o utilizador queira realizar. De notar que estamos a considerar actualmente apenas agrupamentos com base em atributos espaciais. Se, porventura, considerarmos agrupamentos com base em atributos não espaciais, outras formas de representação mais elaboradas teriam que ser utilizadas. No caso dos polígonos, a união destes não é uma possibilidade. Uma possível forma de representação seria atribuir a mesma cor a "polígonos" que se encontrassem no mesmo grupo.

3.4 Estilos e Legenda

Enquanto os atributos espaciais dão resposta às perguntas como *o quê?* e *onde?* para os dados representados, o *estilo* consiste na descrição de como representar visualmente os dados associados aos atributos espaciais.

Bédard refere que *"a user of a SOLAP tool should be able to modify the graphical semiology according to its specific needs, in order to highlight relevant information.* (…) *The legend should be modifiable* (…) *This would facilitate reclassifying for instance, or highlighting categories*." [**4**].

Deste modo, o potencial que é possível obter da interpretação dos mapas que permitam análises intuitivas e adequadas ao utilizador está dependente, não só do tipo e da definição correcta do estilo, como também da possibilidade de uma legenda modificável.

Assim, para obter estilos correctamente definidos, estes devem de estar estipulados segundo determinadas regras. Uma definição errada de um estilo poderá levar a interpretações difíceis ou completamente erradas dos dados em causa.

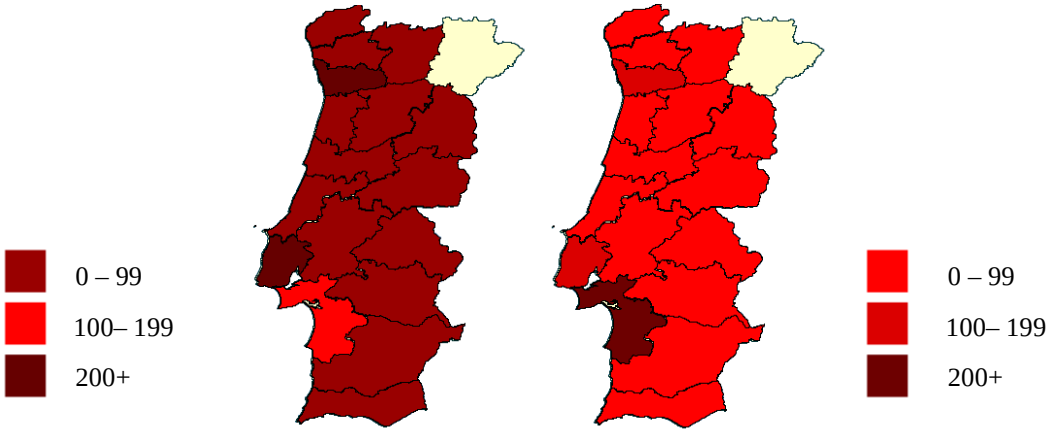


Figura 61 - Ilustra um estilo incorrecto (esquerda) e um estilo correcto (direita).

Na Figura 61, o estilo escolhido para mapear a soma da quantidade de poluição emitida pelas indústrias é a luminosidade como base numa cor inicial: o vermelho. A análise em concreto de um determinado poluente encontra-se ao nível de distritos de Portugal, onde a cada distrito é atribuída uma determinada luminosidade conforme o valor da respectiva métrica (soma da quantidade de poluição).

No mapa da esquerda, o valor da luminosidade não está ordenado consoante os valores das classes. O facto da métrica numérica estar representada por uma variável visual desordenada poderá levar a conclusões erradas. Todavia, mesmo que não provoque uma análise errada, obriga o utilizador a "ler" o mapa, isto é, este é obrigado, para cada classe de valores, a memorizar o valor da variável visual (memorizar a legenda), que por sua vez o leva a reconstruir mentalmente o mapa. Submeter o utilizador a este raciocínio faz com que não se tire partido da superioridade das imagens

face à informação alfanumérica, que se encontra na sua apreensão global e imediata [**34**] permitindo ao utilizador memorizar a imagem automaticamente e interpretar os dados associados a esta.

Relativamente ao mapa da direita, o mapeamento do valor da variável visual respeita a ordem dos dados, isto é, quanto maior a soma da quantidade de poluição, menor é o valor de luminosidade. Como é observável, facilmente se conclui em que distritos se verificam emissões elevadas.

Em consequência, uma variável visual deve expressar a lógica do significado dos dados mapeados. De modo a conceber estilos para que possam ser "vistos" e não "lidos" é necessário aplicar a metodologia da semiologia gráfica desenvolvida por BERTIN [**35**]. Em suma, a aplicação da semiologia gráfica trata-se de uma tradução dos valores alfanuméricos para propriedades gráficas, segundo determinadas regras que serão revistas na secção seguinte (secção 3.4.1).

3.4.1 Semiologia Gráfica

BERTIN reconheceu as seguintes variáveis visuais: tamanho, luminosidade, cor, forma e textura. Estas variáveis visuais podem ser aplicadas em pontos, linhas ou polígonos e são designadas de modos de aplicação em semiologia gráfica.

Níveis de Organização		
Selectivo	**Ordenado**	**Quantitativo**
Forma		
Cor		
	Luminosidade	
	Tamanho	Tamanho
Textura		

Tabela 4 - Níveis de organização para cada variável visual.

Cada variável visual traduz um ou mais significados. Esses significados denominam-se de níveis de organização. BERTIN atribuíu a cada variável visual os significados mais adequados que estas trespassam (Tabela 4). Por exemplo, BERTIN considera que a variável *tamanho* tende a trespassar uma ordenação ou quantidade, mas o mesmo não se verifica para a cor, que está associada a significados selectivos (ex: cores do semáforo).

Por outro lado, segundo a Estatística, os dados podem ser categorizados em diferentes classes. A principal distinção é: dados contínuos e dados discretos. Quanto aos primeiros, os dados podem tomar qualquer valor numérico. Um bom exemplo deste tipo de dados são as métricas numéricas. Ao

contrário dos dados contínuos, os dados discretos apenas podem tomar valores pertencentes a um conjunto finito de valores. Este tipo de dados pode ainda ser subdividido em duas categorias: (*i*) nominal; (*ii*) ordinal. O que difere da categoria nominal para a categoria ordinal é a ausência ou não de uma ordem presente nos dados. Por exemplo, o atributo sexo (masculino ou feminino) é um exemplo de um tipo de dados nominal, pois não existe uma ordem implícita. O mesmo não se verifica para dados ordinais, onde os valores podem ser colocados segundo uma determinada ordem, como por exemplo o número de filhos de um casal.

Com base no significado inerente a cada variável visual e no tipo de dados, foi determinado se uma variável visual era apropriada, ou não, para mapear os dados num determinado modo de aplicação. Na tabela seguinte (Tabela 5 retirada de [**36**]) foi determinado em que contexto é adequado utilizar cada uma das variáveis visuais.

Tipo de Objecto Espacial	Tipo de dados	Tamanho	Luminosidade	Cor	Forma	Textura
Ponto	**Nominal**			√	√	√
	Ordinal	√	√			
	Quantitativo	√	√			
Linha	**Nominal**			√	√	√
	Ordinal	√	√	√ (depende)		
	Quantitativo	√	√	√ (depende)		
Polígono	**Nominal**			√		√
	Ordinal		√	√ (depende)		
	Quantitativo		√	√ (depende)		

Tabela 5 - Variáveis visuais em função do tipo de dados e do modo de aplicação.

A aplicação da variável *cor* tem algumas restrições quando aplicada à linha ou ao polígono. Apesar da variável *cor* ter intrínseco um significado selectivo, se esta for utilizada com algum critério pode, inclusive, mapear dados ordinais e quantitativos quando utilizada em linhas ou polígonos. Se a variável for usada em termos de variação da saturação, então é possível visualmente ilustrar a evolução de dados quantitativos ou ordinais. Caso não haja qualquer critério nas cores utilizadas para mapear os dados, esta variável apenas é apropriada em tipos de dados nominais.

Ao desenvolver estilos mediante as indicações estabelecidas pela Tabela 5, tira-se verdadeiramente partido do potencial das imagens possibilitando ao utilizador realizar análises com menos esforço e mais imediatas.

Porém, BERTIN apenas antevia a aplicação das diferentes variáveis visuais aos objectos espaciais. Relembro que num ambiente SOLAP em muitos casos a informação das métricas é apresentada com recurso a gráficos. Inclusivamente o recurso a estes permite manter a relação de 1:1 entre a tabela de suporte e o mapa em alguns casos de interacção previstos no modelo genérico [**1**]. Assim, é introduzido um novo modo de aplicação das variáveis visuais que visa aplicar as variáveis reconhecidas por BERTIN às características visuais dos gráficos (Tabela 6).

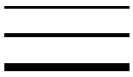
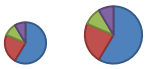
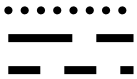
Variável Visual	Ponto	Linha	Polígono	Gráfico de Barras	Gráfico Circular
Tamanho			Não tem aplicação		
Luminosidade					
Cor					
Forma			Não tem aplicação	Não tem aplicação	Não tem aplicação
Textura				Não tem aplicação	Não tem aplicação

Tabela 6 - Variáveis visuais e o respectivo modo de aplicação.

O modo de aplicação da variável *tamanho* ao gráfico de barras incide apenas na espessura das barras. Hipoteticamente, ao variar, não só a largura das barras, como a altura, tornava-se mais complicado, para o analista, comparar os diferentes gráficos de barras associados ao mapa.

Relativamente ao gráfico circular, não é possível utilizar isoladamente a propriedade luminosidade ou a cor, pois, ao utilizar uma destas propriedades, não é possível distinguir os diferentes sectores do gráfico. Assim, a variável luminosidade completa a variável cor da seguinte forma: utiliza-se a cor (para cada gráfico) e de seguida a luminosidade (para cada sector). A aplicação inversa já não é possível.

Por fim, BERTIN afirma também que a utilização de até três destas variáveis visuais em simultâneo permite ao utilizador observar o mapa como uma única imagem [**36**]. Deste modo, na próxima secção (secção 3.4.2) será definido um modelo de estilos com base nas regras da semiologia gráfica revistas anteriormente e nunca ultrapassando o uso de três das variáveis visuais em simultâneo.

3.4.2 Modelo de Estilos

Em [**1**] é sustentado que os estilos aplicáveis dependem de dois principais factores: (*i*) tipo de objecto espacial; (*ii*) número de colunas numéricas. Embora verdade, é uma visão um pouco limitada. O modelo definido em [**1**] não considera a possibilidade de estarem presentes atributos semânticos, não prevê o objecto espacial linha, os estilos definidos como aplicáveis para cada caso poderiam ser mais diversificados e não propõe estilos para cenários de interacção com dois objectos espaciais de diferentes dimensões.

Deste modo, além do tipo de objecto espacial e do número de colunas numéricas, os estilos aplicáveis dependem também do número de objectos espaciais e do número de colunas alfanuméricas. Para cada coluna alfanumérica depende ainda do tipo de dados que a respectiva coluna representa. Assim, os nós de decisão (que definem um contexto) para determinar os estilos possíveis são os seguintes:

Figura 62 - Nós de decisão utilizados no Modelo de Estilos.

A definição do modelo de estilos tem por base o seguinte conjunto de definições:

- **Estilo Simples:** consiste numa única variável visual;
- **Estilo Composto:** consiste na utilização de: (*i*) dois ou mais estilos simples; (*ii*) na combinação entre estilos simples com gráficos; (*iii*) na composição de estilos compostos com qualquer outro estilo.
- **Gráfico:** consiste num gráfico de barras, circular ou um outro qualquer tipo de gráfico.

Sempre que se utilize o estilo composto não deve ser excedida a utilização de três variáveis visuais. De notar que em cada situação uma variável visual é utilizada uma única vez. Adicionalmente às definições anteriores, o modelo de estilos está definido com base na representação da tabela de suporte (Figura 63).

Colunas Alfanuméricas		**Colunas Numéricas**	
		C	
A_1	A_i	C_1	C_i

Figura 63 - Representação da Tabela de Suporte.

O termo *coluna alfanumérica* refere-se ao atributo semântico da dimensão espacial a um nível superior, comparativamente ao atributo espacial, e as colunas numéricas resultam das métricas e dos atributos semânticos da dimensão espacial a um nível inferior ou dos atributos semânticos de dimensões semânticas. Através destes conceitos, restrições e organização da tabela de suporte são definidos os modelos de estilos que descrevem quais os estilos possíveis para cada contexto.

Em casos em que se está na presença de **um objecto espacial** e este corresponde ao objecto geográfico **ponto** é definido o seguinte modelo quando apenas temos uma coluna numérica:

- **Número de Colunas Alfanuméricas = 0:**

  A coluna numérica pode ser mapeada pelas variáveis visuais *Tamanho* ou *Luminosidade*. Outra possibilidade é a utilização de um gráfico de barras.

- **Número de Colunas Alfanuméricas = 1:**

  - **Tipo de Dados = Nominal:**

    Na presença de uma coluna numérica e uma coluna alfanumérica, é necessário um estilo composto com dois "argumentos", um para cada coluna. Para a coluna numérica os estilos indicados são os mesmos que para quando o número de colunas alfanuméricas é zero. Para a coluna alfanumérica, os estilos devem ser mapeados por variáveis com significado selectivo. São elas: *Forma*, *Cor* e *Textura*.

  - **Tipo de Dados = Ordinal:**

    Ao contrário do contexto anterior, neste contexto a coluna alfanumérica é do tipo ordinal. Assim, apenas os estilos para a coluna alfanumérica são alterados. Deste modo, os *estilos simples* possíveis para a coluna alfanumérica são: *Tamanho* e *Luminosidade*. De notar que o estilo composto não pode conter estilos repetidos.

- **Número de Colunas Alfanuméricas $\geq$ 2 e $\leq$ 3:**

  Para este contexto podem existir duas formas de estilo composto. Um estilo que combina apenas variáveis visuais (*Estilo Composto 1*) e outro estilo que combina o gráfico de barras (para a coluna numérica) com variáveis visuais para as colunas alfanuméricas. Para cada coluna, as variáveis visuais possíveis, que este modelo considera, seguem as mesmas restrições que nos contextos anteriores.

No caso em que se tem até três colunas alfanuméricas, é necessário realizar alguns comentários. Em primeiro lugar, o *Estilo Composto 1* apenas pode ser utilizado se se verificar apenas duas colunas

alfanuméricas. Se assim não fosse, teríamos que fazer uso de quatro variáveis visuais em simultâneo e perdia-se o potencial das imagens.

Em [**1**] por vezes era argumentado que para cada situação existia apenas um estilo recomendado. Apesar de existirem algumas situações particulares, em que claramente existe um estilo que se adequa mais perante qualquer outro, na verdade é que nem sempre existe um estilo indiscutivelmente melhor entre os aplicáveis.

Por vezes, diferentes estilos têm diferentes propósitos e podem destacar diferentes informações. Considere o exemplo sobre emissões de poluentes das indústrias em Portugal (ao nível das indústrias). Se o utilizador quiser analisar quais as indústrias que poluem mais ou menos, ou porventura analisar "zonas" com instalações muito poluidoras, então a utilização de uma das propriedades visuais é uma boa escolha. No entanto, se um utilizador necessitar de efectuar uma análise mais minuciosa e quiser comparar os valores de emissão entre as diversas indústrias, o gráfico de barras torna-se uma melhor escolha. Através das propriedades visuais não seria possível comparar valores numéricos entre indústrias que tivessem associado o mesmo valor da propriedade visual.

No entanto, no modelo anterior apenas se define parte da árvore de decisão. Em casos com duas colunas numéricas o modelo de estilos é o seguinte:

- **Número de Colunas Alfanuméricas = 0:**

  Com duas colunas numéricas, estas podem ser interpretadas por duas variáveis visuais: *Tamanho* e *Luminosidade*. Outra possibilidade é a utilização de um *Gráfico* (de barras ou circular).

- **Número de Colunas Alfanuméricas = 1:**

  - **Tipo de Dados = Nominal:**

    Neste contexto faz sentido a utilização de um estilo composto que aplica a variável *Cor* sobre o gráfico de barras ou circular. Neste estilo, o gráfico circular ou de barras mapeia as duas colunas numéricas e a *Cor* interpreta a coluna alfanumérica. Um outro tipo de estilo é um estilo que combina apenas variáveis visuais, não esquecendo as devidas restrições explicitadas na Tabela 5.

- **Tipo de Dados = Ordinal:**

  Se no contexto anterior era utilizada a *Cor* sobre o gráfico de barras ou circular, pelo facto da coluna numérica ser nominal, neste caso a abordagem é semelhante, mas em vez de se aplicar a variável *Cor*, aplica-se a variável *Tamanho* ou *Luminosidade* (apenas no gráfico de barras).

- **Número de Colunas Alfanuméricas ≥ 2 e ≤ 3:**

  Quando se tem mais do que duas colunas alfanuméricas, a utilização de estilos que combinem apenas variáveis visuais não é possível. Os estilos possíveis não são infinitos, e, para manter as regras revistas anteriormente só é possível definir estilos enquanto é possível manter as regras da Tabela 5 e Tabela 6. Por exemplo, na presença de duas colunas alfanuméricas só é possível utilizar o *Gráfico Circular* se uma das colunas for do tipo nominal e outra ordinal (combinando-se a aplicação da variável *Tamanho* com a *Cor* sobre o gráfico circular).

No seguinte exemplo (Figura 64) encontra-se um caso em que se pretende avaliar qual é a relação de vendas/categoria de produto para duas categorias distintas. Os valores apresentados na tabela representam a soma das vendas.

		Categoria de Produto	
	Tipo de Loja	**Tipo1**	**Tipo2**
Loja 1	Hipermercado	619.7	536.16
Loja 2	Supermercado	60.64	553.80
Loja 3	Hipermercado	301.37	671.58
Loja 4	Supermercado	80,94	525.08
Loja 5	Supermercado	599.67	457.32
Loja 6	Hipermercado	520.70	795.63

Figura 64 - Dados não reais num contexto de vendas.

Dado que o tipo de dados do atributo *tipo de loja* é nominal e se está na presença de duas colunas numéricas, a utilização do *Estilo Composto* (*Cor*, *Gráfico*) resulta no seguinte mapa (Figura 65):

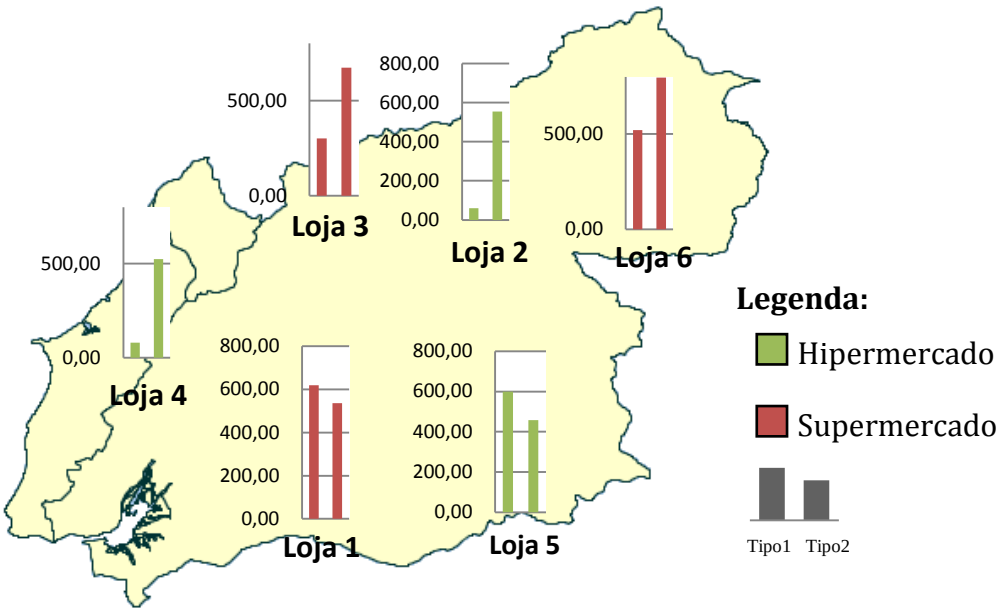


Figura 65 - *Estilo Composto* (*Cor, Gráfico de Barras*) aplicado ao contexto de vendas.

Através do estilo utilizado, para o conjunto de dados muito elementar, é possível retirar que os hipermercados, em geral, têm um número de vendas superior que os supermercados em ambos as categorias de produtos. É possível observar que a loja 5 tem um comportamento divergente relativamente às vendas dos outros supermercados para a categoria *tipo1*.

Com este pequeno exemplo é visível que este estilo é uma boa escolha quando o utilizador pretende analisar a relação das entidades com os mesmos ou diferentes valores do atributo semântico e descobrir *outliers* ou tendências, tendo em conta os atributos semânticos e métricas numéricas em análise.

Além dos estilos que combinam as variáveis visuais com gráficos para os contextos de interacção previstos com duas colunas numéricas, existem estilos que apenas combinam variáveis visuais. Independentemente do caso de interacção, estes estilos são adequados quando o utilizador pretende procurar por entidades semelhantes, descobrir a existência de alguma correlação entre a relação espacial (adjacência, conectividade, proximidade) ou distribuição espacial (concentrado, disperso) das localizações geográficas das entidades e os dados semânticos em análise.

Por fim, quando surgem casos em que o número de colunas numéricas é superior a dois, os estilos possíveis são os seguintes:

- **Número de Colunas Alfanuméricas = 0:**

  Para este contexto, os estilos possíveis são apenas do tipo *Gráfico*.

- **Número de Colunas Alfanuméricas = 1:**

  O tipo de estilo possível para este contexto foi já apresentado anteriormente, quando se introduz o modelo de estilos para duas colunas numéricas. Esse corresponde ao estilo que utiliza uma variável visual sobre o tipo *Gráfico*, mantendo as regras da Tabela 5 e Tabela 6.

- **Número de Colunas Alfanuméricas ≥ 2 e ≤ 3:**

  Neste contexto os estilos possíveis correspondem apenas àqueles que combinam variáveis visuais (para interpretar as colunas alfanuméricas) com o tipo *Gráfico* (que apresenta as colunas numéricas). Novamente, as regras da Tabela 5 e Tabela 6 são mantidas nas diversas combinações possíveis.

Com o aumentar do número de colunas numéricas, os estilos possíveis são bastante reduzidos.

O aumento do número de colunas numéricas apenas implica um aumento de número de barras ou partições, conforme se esteja na presença de um gráfico de barras ou circular. Por isso, apenas se mantêm os estilos que combinam as propriedades visuais com os gráficos, relativamente ao modelo que prevê duas colunas numéricas.

Na ausência de colunas alfanuméricas apenas é possível utilizar gráficos para representar as colunas numéricas, pelo facto de existirem apenas duas propriedades que trespassam a noção de ordenação (luminosidade e tamanho).

Os modelos de estilos para a linha e para o polígono seguem a mesma base de desenvolvimento. Através da Tabela 6 as variáveis visuais têm diferentes modos de aplicação. Este facto contribui para que algumas das variáveis visuais utilizadas no modelo de estilos dos pontos não sejam usadas nos modelos de estilos das linhas e dos polígonos. Outras questões se levantam na construção dos respectivos modelos de estilos.

O modelo de estilos que prevê casos com dois atributos espaciais de diferentes dimensões difere um pouco dos anteriores, pois tem que equacionar que o estilo pode também envolver as extremidades. A continuação da definição do modelo de estilos para a linha, polígono e para os casos com dois atributos espaciais de diferentes dimensões encontra-se em anexo (secção A.1).

3.4.3 Gestor de Estilos

O objectivo da definição do gestor de estilos é possibilitar a alteração de estilos e permitir uma legenda alterável pelo utilizador, verificando sempre uma definição correcta dos estilos.

Deste modo, o gestor de estilos é a entidade que proporcionará a alteração/criação de estilos e dará ao utilizador a capacidade de modificar a legenda. Assim, o modelo para suportar a gestão, alteração e criação dos estilos é o apresentado na Figura 66.

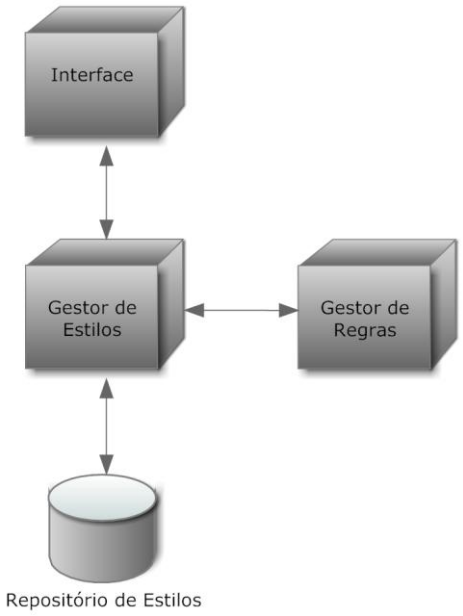


Figura 66 - *Framework* para a gestão de estilos.

O modelo para a gestão dos estilos é composto por quatro componentes: (i) interface; (ii) Gestor de Estilos; (iii) Gestor de Regras; (iv) Repositórios de Estilos.

É através da **interface** que o utilizador irá interagir para criar/alterar estilos e modificar a respectiva legenda. Assim, a interface é um componente que deverá estar integrado com a interface oferecida ao utilizador. Por outro lado, este componente deverá estar completamente desacoplado da lógica da gestão dos estilos.

Antes de mais é necessário introduzir dois conceitos: (*i*) tipo de estilo; (*ii*) instância de estilo. O tipo de estilo consiste na definição da natureza do estilo. Por exemplo, um tipo de estilo pode ser *Estilo Simples (Tamanho).* A instância de estilo consiste na definição concreta de um estilo, isto é, definição dos atributos que compõem um estilo. Por exemplo, para a propriedade *Tamanho* a definição concreta poderia ser:

- Propriedades Visuais (ex: cor)
- Número de Intervalos
- Tipo de mapeamento: Uniforme ou Personalizado

Quanto ao conceito *contexto*, este é definido pelos nós de decisão referidos na secção 3.4.2.

Assim, como base nestes dois conceitos, o **gestor de estilos** suporta duas primitivas: (*i*) obter tipos de estilos dado um contexto; (*ii*) obter instâncias de estilos dado um tipo de estilo.

Estas duas primitivas são a base de comunicação com a interface. É a partir destas duas primitivas que é dado suporte ao utilizador para que este, em tempo de análise, possa alterar não só o tipo de estilo como também escolher, dentro do mesmo tipo de estilo, instâncias de estilos com diferentes definições concretas.

Todavia, dar liberdade ao utilizador para alterar e criar os seus próprios estilos poderá levar ao conflito das regras da semiologia gráfica revistas anteriormente. Para que esta liberdade não prejudique as análises do utilizador, é introduzido o componente **gestor de regras.**

O gestor de regras suporta uma primitiva, designada de estilos aplicáveis. É através desta primitiva que o gestor obtém os tipos de estilos aplicáveis dado um contexto de análise, eliminando os possíveis conflitos que pudessem vir a ser provocados pelo utilizador.

O papel fundamental do modelo de estilos referido anteriormente é o de eliminar possíveis conflitos e para cada contexto oferecer os estilos aplicáveis. Com isto, o gestor de regras encapsula todo o conhecimento de estilos aplicáveis a um dado contexto, permitindo ao gestor de estilos retornar ao utilizador apenas estilos benéficos para as análises.

O **repositório de estilos** tem como propósito guardar a definição concreta dos diversos estilos. A partir do momento em que o gestor de estilos contém informação do tipo de estilos aplicáveis ao contexto de análise do utilizador, este tem que recorrer ao repositório de estilos para obter a definição concreta destes. Após a escolha do utilizador, a instanciação do estilo é realizada com base na respectiva definição concreta.

Por fim, o suporte para permitir uma legenda modificável está implícito na solução, isto é, sempre que o utilizador queira alterar a legenda apenas é necessário alterar a definição concreta do estilo e voltar a instanciar.

Todos os assuntos introduzidos e discutidos nesta secção em redor dos estilos, modelos de estilos e gestor de estilos são uma base de trabalho, com o objectivo de introduzir nos sistemas SOLAP uma gestão adequada dos estilos, permitindo adquirir do mapa resultados correctos e efectivos.

Capítulo 4

Arquitectura

Este capítulo apresenta a arquitectura geral do sistema SOLAP+, seus componentes, modelos e protocolo de comunicação.

Este capítulo apresenta a arquitectura do protótipo SOLAP. Começa por apresentar a arquitectura conceptual do sistema, detalhando de seguida cada um dos seus componentes. Apresenta o protocolo de comunicação usado entre o cliente e o servidor e descreve o meta-modelo que dá suporte ao sistema.

A arquitectura do sistema baseia-se na arquitectura desenvolvida pelo Ruben Jorge [**1**] e são seguidos os princípios da mesma. Pelo facto de se querer estender o sistema SOLAP+, existe a necessidade de introduzir alguns componentes e realizar alguns acertos na arquitectura anterior.

4.1 Arquitectura Geral

Existem cinco componentes na arquitectura do protótipo: (*i*) cliente; (*ii*) servidor; (*iii*) SGBD; (*iv*) servidor de mapas; (*v*) repositório de metadados.

O **cliente** é responsável por toda a interacção com o utilizador, apresentação de dados e pela geração do pedido para o servidor. É também responsável por efectuar o pedido ao **servidor de mapas** com o objectivo de apresentar o mapa ao utilizador. O **servidor** tem como função receber os pedidos dos clientes, processá-los utilizando os **metadados** necessários e comunicar com o **SGBD**. O **servidor de mapas** é uma componente externa que permite a geração dinâmica de mapas temáticos.

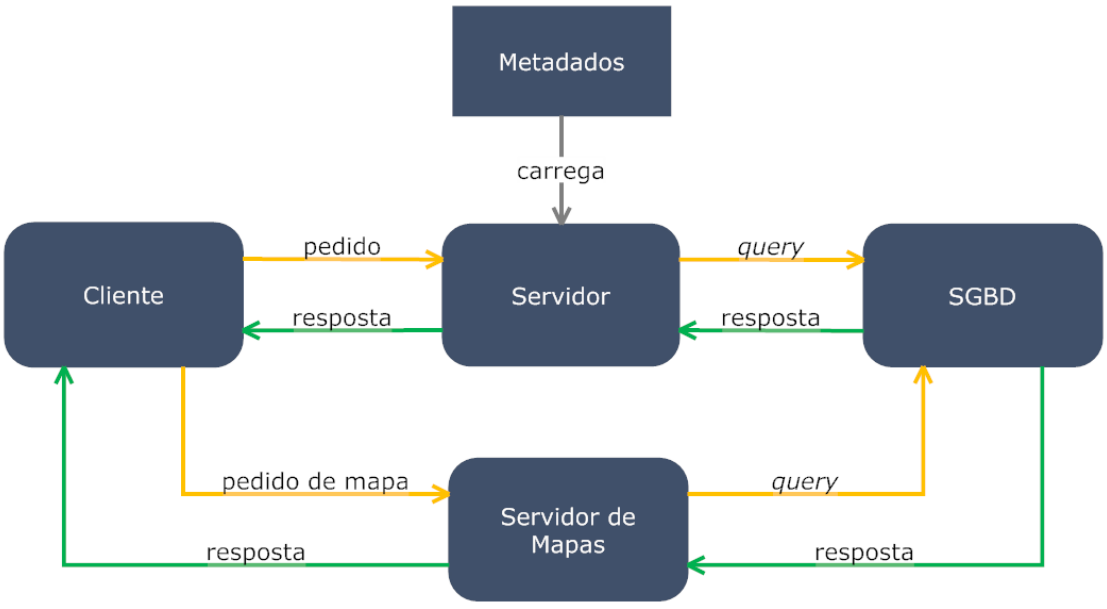


Figura 67 - Arquitectura geral do protótipo.

4.2 Servidor

Quando um pedido é submetido ao servidor, este é de imediato sujeito a validação. A componente responsável por efectuar a validação é o **módulo de comunicação.** A validação é realizada a partir de um *XML Schema* interno.

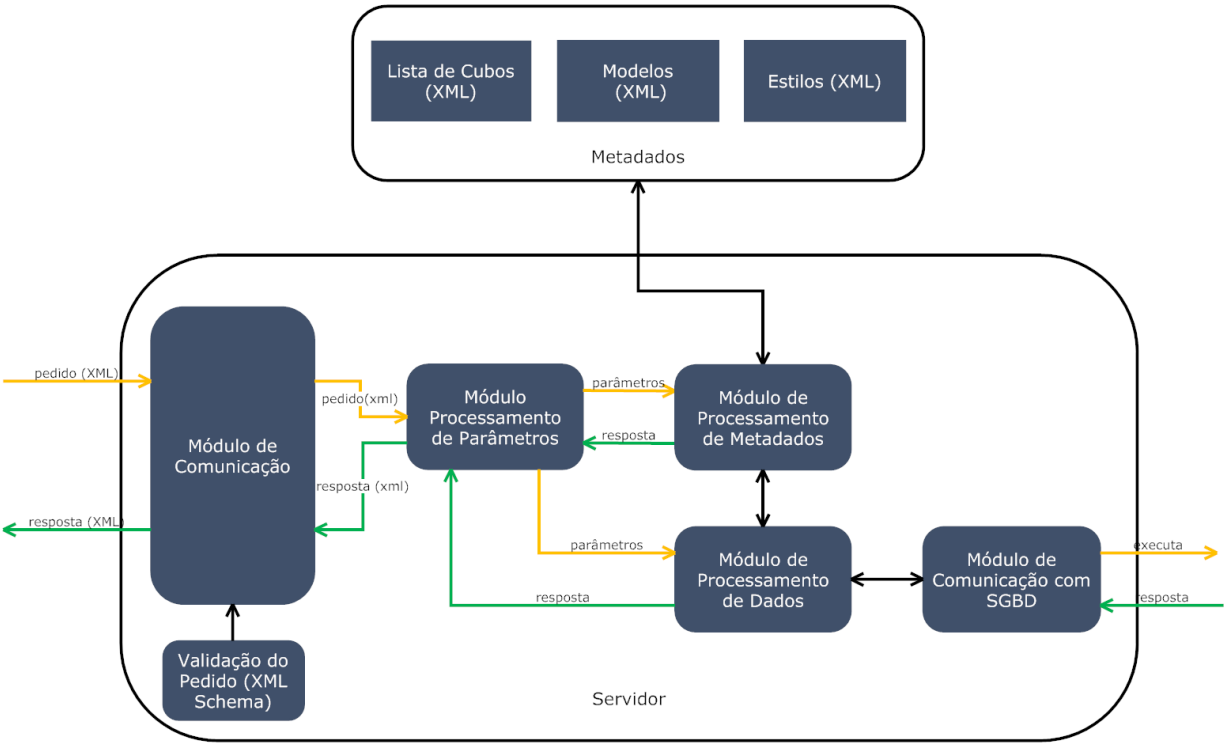


Figura 68 - Arquitectura do Servidor.

Assumindo que o pedido está correctamente definido, este é enviado para o **módulo de processamento de parâmetros**. Este componente tem a responsabilidade de realizar o *parser* do pedido e extrair todos os parâmetros associados. Os parâmetros são enviados para o próximo componente apropriado, consoante o tipo de pedido.

O **módulo de processamento de metadados** dá resposta aos pedidos que não necessitam de acesso à base de dados. Esses pedidos englobam apenas informação que está contida nos **metadados** (como por exemplo qual a lista dos modelos de dados) e, portanto, esta componente é auto-suficiente para produzir uma resposta.

O mesmo não se verifica quando o pedido envolve acesso à base de dados. O componente responsável por lidar com estes pedidos é o **módulo de processamento de dados** (detalhado abaixo, ver Figura 69)**.** Este componente interage com o **módulo de comunicação com SGBD** que tem como função receber as interrogações de outros componentes e submetê-las ao SGBD. Se for necessário, retorna os resultados das interrogações. Assim que o módulo de processamento de dados

tiver produzido a resposta, esta é encaminhada para o **módulo de processamento de parâmetros**, no qual é produzida a resposta em XML. Por último, é enviada a resposta para o cliente através do **módulo de comunicação.**

O **módulo de processamento de dados** (Figura 69) é o componente mais importante e, como tal, o mais complexo na arquitectura do servidor. Inicialmente, com base nos parâmetros e nos metadados que descrevem o modelo multi-dimensional, o **navegador de agregados** escolhe o agregado apropriado (em [**1**] é feita a descrição com mais detalhe de como é feita a sua escolha). Após a escolha do agregado, é remetido para o **gerador *sql*** a informação do agregado escolhido, como também os respectivos parâmetros. O **gerador *sql*** pode ou não utilizar os componentes auxiliares (ex: gerador de tabelas), conforme o tipo de pedido. Quando a resposta estiver pronta, esta é enviada para o **módulo de processamento de parâmetros.**

O **gerador da *query* espacial** tem unicamente a função de produzir uma *query*. Esta será posteriormente anexada na resposta. O cliente utilizará esta *query* para a encaminhar ao **servidor de mapas**, de modo a obter o mapa com a mesma informação que está presente na tabela de suporte.

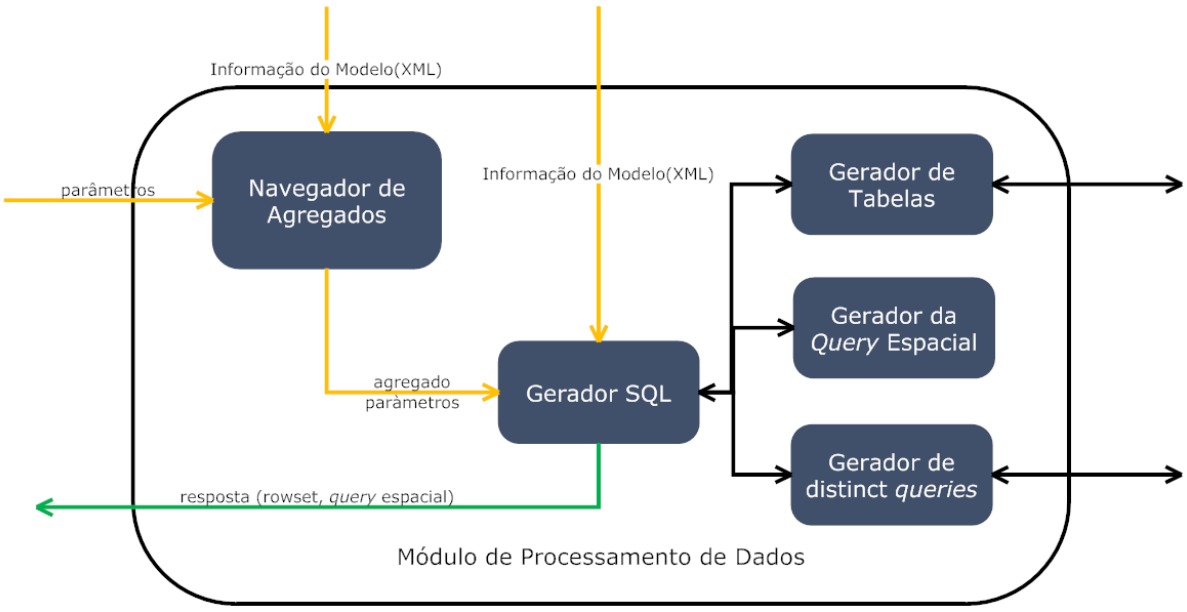


Figura 69 - Módulo de Processamento de Dados na arquitectura do servidor.

Em alguns casos, a *query* gerada com base nas entidades originais do modelo multi-dimensional (tabelas de facto e dimensões) não é apropriada para ser enviada ao **servidor de mapas**: ou porque são necessárias novas representações espaciais; ou porque o conjunto de linhas resultante não respeita as restrições da tabela de suporte; ou, em última instância, ambos os casos.

De modo a adaptar a arquitectura anterior [**1**] para suportar casos onde existe a necessidade de computar novas representações espaciais, foi concebido o seguinte **gerador de tabelas** (Figura 70):

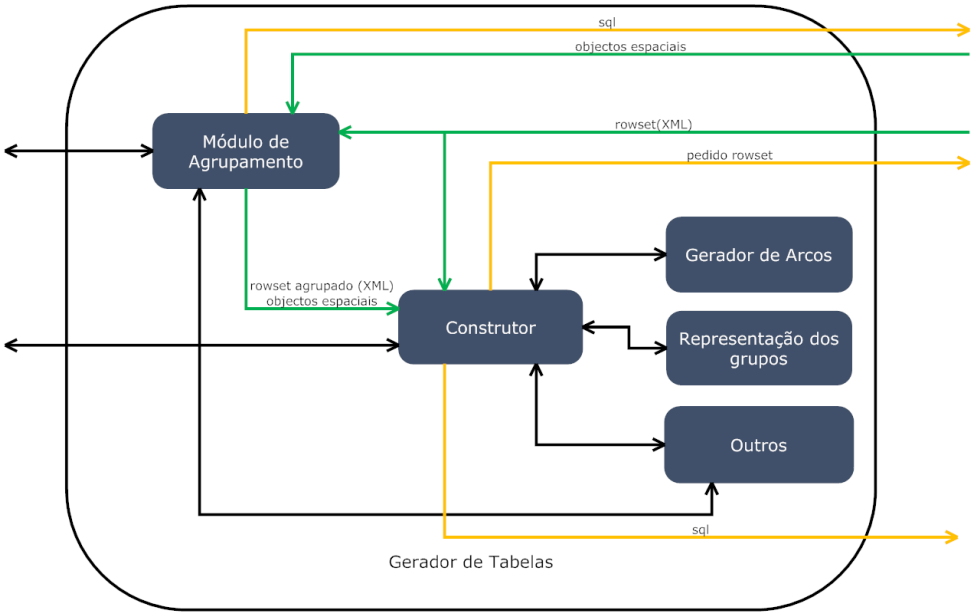


Figura 70 - Gerador de Tabelas.

Inicialmente, o **construtor** é responsável por enviar a *query* gerada com base nas entidades originais. Consoante os parâmetros do pedido, o *rowset* resultante é sujeito, ou não, ao **módulo de agrupamento**. Este componente engloba os algoritmos de agrupamento espacial e é responsável pela agregação dos dados de cada grupo, conhecendo *a priori* o operador de agregação. Após o **construtor** receber o *rowset* (tenha este percorrido a fase de pré-processamento ou não) este é responsável, se for necessário, por o colocar em forma pivô. Se forem necessárias novas representações espaciais dos objectos, então o(s) gerador(es) das novas representações espaciais a utilizar irá depender dos parâmetros do pedido. Na presença de agrupamento espacial é utilizado o **gerador de representação de grupos.** Com dois atributos espaciais de diferentes dimensões é utilizado o **gerador de arcos**. Quando as representações estiverem prontas, é gerada e preenchida a tabela com os respectivos dados organizados respeitando a restrições da tabela de suporte (ver Figura 63).

4.3 Cliente

Através da interacção do utilizador, a componente **interface** é responsável por despoletar os pedidos para o servidor. À semelhança do que se verificava na arquitectura do servidor, no cliente existe também o **módulo de processamento de parâmetros** que é responsável tanto por receber os

parâmetros dos pedidos e produzir um XML a ser enviado para o servidor, como por receber a resposta do servidor e extrair os parâmetros associados a esta.

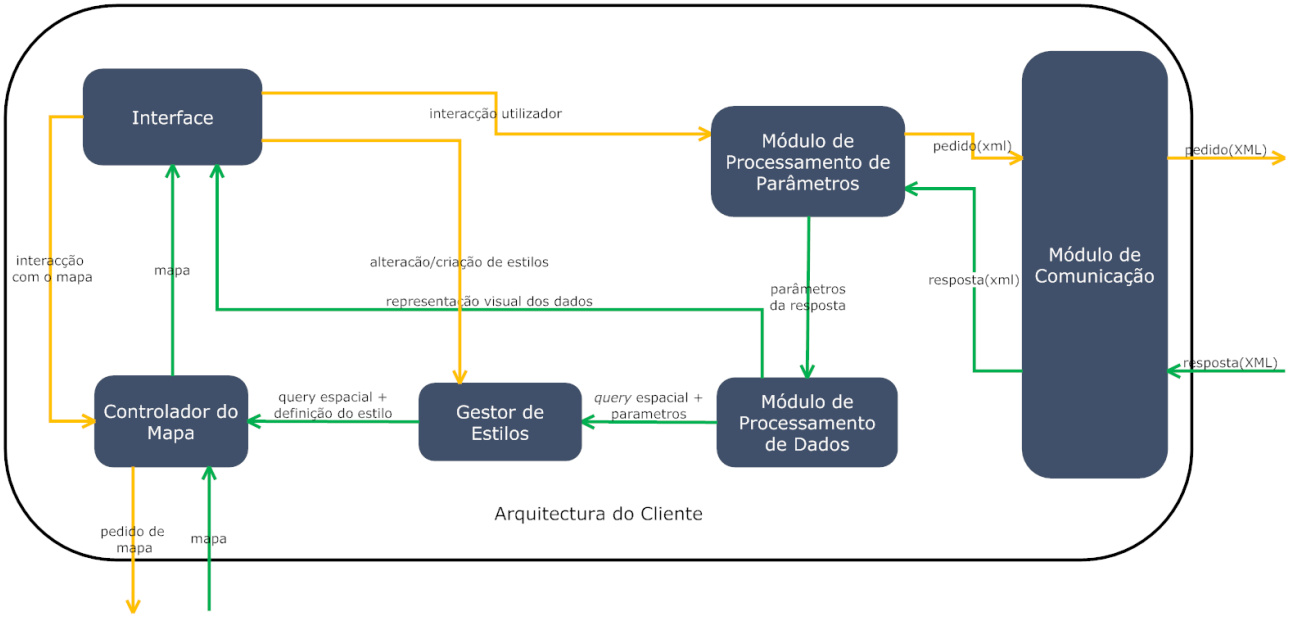


Figura 71 - Arquitectura do Cliente.

Após serem extraídos os parâmetros de uma resposta, estes são encaminhados para o **módulo de processamento de dados** (Figura 72). Este é constituído por vários componentes que têm como função produzir representações visuais dos dados recebidos com base nos parâmetros da resposta. Estas representações visuais são enviadas para a **interface** onde são apresentadas.

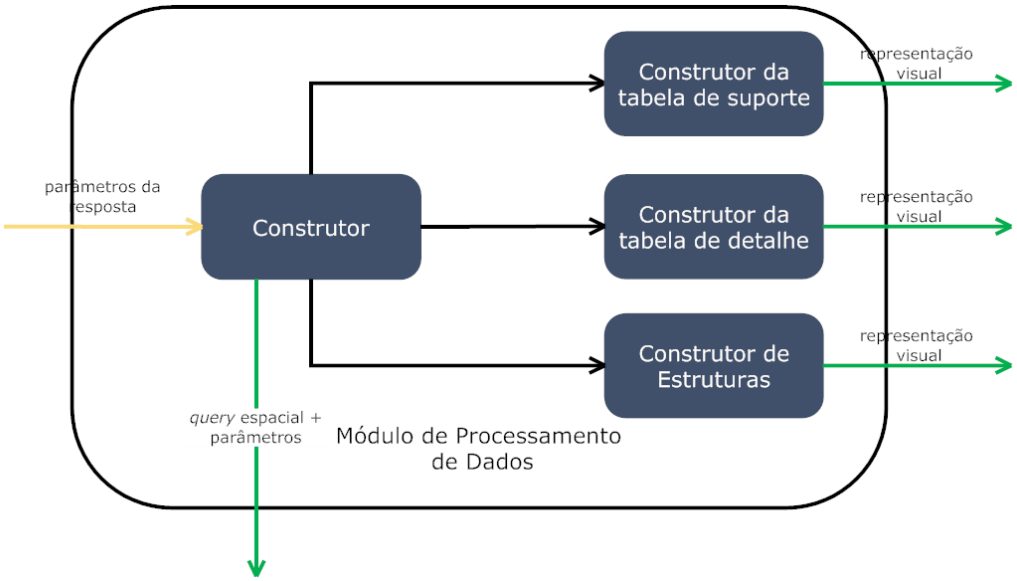


Figura 72 - Módulo de Processamento de Dados na arquitectura do cliente.

Num típico pedido para obter dados, as diversas representações visuais utilizadas são a tabela de suporte e/ou tabela de detalhe, que permitem gerar as representações visuais para o conjunto de dados recebido. O **construtor de estruturas** é utilizado quando os pedidos (ex: carregamento do cubo) requerem árvores, listas ou outros componentes que estão associados a estruturas de dados.

O **construtor** é também responsável por encaminhar a *query* espacial e os parâmetros da resposta para o **gestor de estilos**.

Este é responsável, com base nos parâmetros da resposta e numa árvore de decisão definida nos metadados do modelo multi-dimensional (detalhado na secção 4.5.1.2), escolher a definição do estilo apropriado que está descrito no repositório de estilos (*Estilos XML*). Produz também o estilo, o que inclui gerar as classes da legenda com base nos dados recebidos e gerar código para a sintaxe do *servidor de mapas*.

Tanto a informação da árvore de decisão, como as definições dos diversos estilos, foram previamente carregadas quando foi realizado o carregamento do cubo. A árvore de decisão baseia-se no modelo de estilos que foi definido na secção 3.4.2.

De seguida, é enviada a *query* espacial com a definição do estilo produzido para o **controlador do mapa**. Com base nesta informação, constrói o pedido ao servidor de mapas e um novo mapa temático é recebido. O **controlador do mapa** é também a "ponte" (entre a **interface** e o **servidor de mapas**) que endereça a interacção que ocorre no mapa (*panning*, visualização da legenda, aumentar/diminuir zoom, etc.).

4.4 Protocolo de Comunicação

O servidor é *stateless* e, como tal, a comunicação entre o cliente e o servidor baseia-se no paradigma pedido - resposta. Todos os pedidos têm a seguinte estrutura base:

```
<solapplus>
    <request call="tipo_de_pedido" spatial="valor_booleano">
      …
    </request>
</solapplus>
```

Os tipos de pedidos podem ser: (*i*) listar cubos; (*ii*) carregar cubo; (*iii*) obter dados; (*iv*) obter distintos. Os primeiros dois tipos de pedidos consistem em apenas obter informação guardada nos metadados. Quanto aos dois últimos, esses já requerem acesso à base de dados. O pedido *obter*

distintos tem como finalidade oferecer ao utilizador a possibilidade de visualizar os valores distintos para um dado atributo das dimensões e, a partir daí, poder seleccionar os valores pelos quais efectua *slices* semânticos. O pedido *obter dados* é detalhado abaixo, pois foi aquele que sofreu pequenas alterações comparativamente com o protocolo definido por Ruben Jorge [**1**]. O atributo *spatial* indica ao servidor se é necessário gerar a *query* espacial ou não.

O pedido *obter dados* segue a estrutura do seguinte exemplo:

```
<params cubeId="1" spatial="true"/>
<measure id="1" operator="SUM" />
<level dimensionId="1" id="2" />
<attribute dimensionId="1" levelId="1" attributeId="1" />
<slice dimensionId="1" levelId="3" attributeId="1" operator="LESS" value1="100" />
<spatialSlice dimensionId="1" levelId="3" attributeId="7" layerId="2" operator="INSIDE" />
<fieldFilter measureId="1" operator="GREATER" value1="2500" />
<measureFilter measureId="1" operator="GREATER" value1="100" measureOperator="SUM"/>
<nFilter measureId="1" operator="TOP" nRows="10" measureOperator="AVG"/>
<clustering mode="AUTO" groups="0" variant="BASED_ON_HIERARCHY" zoomLevel="4"
    <parameters dimensionId="1" levelId="2" hierarchyId="2"/>
</clustering>
```

Os elementos *params*, *measures*, *level*, *attribute*, *slice*, *spatialSlice*, *fieldFilter*, *measureFilter*, *nFilter* e *clustering* descrevem, numa visão genérica, a análise que se quer realizar e é a partir desta informação que são geradas as *queries* no servidor para obter os dados.

Ao protocolo concebido anteriormente [**1**] foi adicionado o elemento *clustering*. O atributo *mode* pode ter como valor *manual* ou *auto*, consoante o utilizador queira, ou não, que o sistema detecte automaticamente se existe a necessidade de aplicar o agrupamento espacial. O atributo *groups* indica o nível com que é realizado o agrupamento (= 0 agrupamento normal; > 0 com mais grupos; < 0 menos grupos). O atributo *variant* indica se se aplica agrupamento *adhoc* ou se se restringe o agrupamento com base num nível de uma hierarquia espacial. A hierarquia escolhida está descrita no elemento *parameters*. O *zoomLevel* indica o nível actual de zoom presente no mapa. Os elementos *parameters* identificam a dimensão e o respectivo nível espacial para o qual se quer realizar o agrupamento.

A resposta a este pedido tem o seguinte formato:

```
 <query sql="..." geometryType="point" />
<table count="…" nMeasures="…">
      <rowset>...</rowset>
       <associatedAttributes>...</associatedAttributes>
      <attributesLevels>...</attributesLevels>
</table>
```

Na resposta vem, por um lado, a informação do conjunto de dados resultante e, por outro, a *query* espacial que será encaminhada para o servidor de mapas. Contém também informação sobre os atributos espaciais (*associatedAttributes*) e os atributos seleccionados (*attributesLevels*). O elemento *attributeLevels* descreve também a relação destes com os atributos espaciais.

4.5 Meta-Modelo

O meta-modelo consiste na descrição de toda a informação necessária pelo modelo genérico SOLAP. A informação necessária inclui a descrição das bases de dados, a informação sobre o servidor de mapas, estilos e modelo multi-dimensional. O meta-modelo é descrito por um ficheiro XML de acordo com a especificação de um *XML Schema*.

Os elementos que se encontram na raiz são: (*i*) *databases*; (*ii*) *mapservers*; (*iii*) *styles*; (*iv*) *multidimensional*. Ao meta-modelo anterior [**1**] foi introduzido o elemento *styles*. Outras alterações foram realizadas nos outros elementos. Apenas serão detalhados os elementos onde houve alterações significativas.

O elemento *database*s contém a descrição das bases de dados relacionais que dão suporte aos modelos multi-dimensionais definidos no elemento *multidimensional*. Para cada base de dados existe um elemento *database*.

O elemento *mapservers* contém a descrição dos servidores de mapas que podem ser utilizados pela componente *mapa*. A descrição inclui: (*i*) a informação necessária para conectar ao servidor; (*ii*) a descrição de *layers* ou objectos espaciais que são utilizados como informação contextual ou para se realizarem *slices* espaciais; (*iii*) a descrição do mapa base. Para cada servidor de mapas existe um elemento *mapserver*. Tanto o elemento *databases* como o *mapservers* estão descritos em pormenor em [**1**].

4.5.1 Elemento *styles*

A introdução do elemento *styles* no meta-modelo veio dar suporte a todo o processo de gestão de estilos. O elemento *styles* contém diversos elementos *style* onde são declarados vários estilos. É também descrito o elemento *decisionTree* que consiste na definição da árvore de decisão. Esta dá suporte ao processo de decisão do estilo a aplicar, que se verifica no componente *gestor de estilos*.

4.5.1.1 Elemento *style*

A estrutura base para a definição de um estilo é a seguinte:

```
<style id="1">
    …
</style>
```

A cada estilo está associado um identificador. Dentro do elemento estilos podem existir diferentes definições. A linguagem para a definição dos estilos tem dois propósitos: (*i*) permitir o desacoplamento entre o sistema e o servidor de mapas (a forma como são definidos os estilos); (*ii*) uma linguagem de alto nível para a definição de estilos, que permitem obter tanto mapas temáticos como mapas com gráficos associados. Assim, um estilo pode ser: (*i*) um atributo visual; (*ii*) um gráfico; (*iii*) um estilo composto.

Com base no que foi analisado na secção Semiologia Gráfica (secção 3.4.1), os atributos visuais previstos na definição de estilos são os seguintes: (*i*) luminosidade (elemento *variableBrightness*); (*ii*) cor (elemento *variableColor*); (*iii*) tamanho (elemento *variableSize*); (*iv*) forma (elemento *variableShape*); (*v*) textura (elemento *variableTexture*).

O elemento *variableBrightness* define um estilo onde é utilizada a luminosidade para traduzir os dados contínuos em propriedades visuais. Este elemento contém informação para a definição da cor base, quer para o contorno, quer para o preenchimento (*strokeColor*, *baseColor*). Para que o estilo possa ser aplicado à forma geométrica *ponto* é necessário também especificar o atributo *marker*. O atributo *typeOfDistribution* define a distribuição dos dados consoante o número de classes definidas no atributo *numberOfClasses*.

```
<variableBrightness marker="STAR" strokeColor="BLACK" baseColor="RED"
                    typeOfDistribution="UNIFORM" numberOfClasses="5" >
    <gradientDefinition/>
    <intervals/>
</variableBrightness>
```

No entanto, o tipo distribuição definido pode não ser uma distribuição pré-definida. De modo a dar suporte a estilos com uma legenda modificável e definida pelo utilizador, são descritos os elementos *gradient definition* e *intervals*, como se pode observar no exemplo seguinte:

```
<variableBrightness marker="STAR" baseColor="RED" strokeColor="BLACK"
                    typeOfDistribution="CUSTOM">
    <gradientDefinition>
        <color id="1" value="#FF0000"/>
        <color id="2" value="#CC0000"/>
    </gradientDefinition>
    <intervals>
        <intervalDefinition lowerBound="4" upperBound="7" label=">= 4 e < 7" propertyID="1"/>
        <intervalDefinition lowerBound="7" upperBound="30" label=">= 7 e < 30" propertyID="2"/>
    </intervals>
</variableBrightness>
```

Neste caso, o elemento *intervals* contém a definição dos limites do intervalo e a respectiva propriedade associada. Este estilo pode também ser utilizado para mapear uma variável discreta ordinal. Nesse caso, em vez de se usar o elemento *intervals*, utiliza-se o elemento *sets*.

```
<sets>
    <setDefinition values="1" label="Um Filho" propertyID="1"/>
    <setDefinition values="2" label="Dois Filhos" propertyID="2"/>
</sets>
```

O elemento *sets* pode conter um ou mais elementos *setDefinition*, onde cada elemento define qual o atributo da variável visual que traduz os valores definidos no atributo *values*.

Os restantes elementos que definem as restantes propriedades visuais seguem a mesma estrutura que o elemento *variableBrightness*, onde apenas difere a designação dos elementos.

Os estilos podem também ser gráficos de barras, circulares ou qualquer outro. No actual meta-modelo prevê-se a definição de estilos associados a gráfico de barras e circulares.

O elemento *barchart* define o estilo para o gráfico de barras. Os atributos *height* e *width* definem a altura e largura das barras, respectivamente. Consoante o valor do atributo *axisXOn*, os gráficos resultantes têm, ou não, definido o eixo dos *xx*. O atributo *globalScale* define uma escala global para todos os gráficos, permitindo ao utilizador comparar os diferentes gráficos.

```
<barchart height="30" width="40" axisXOn="false" globalScale="true"  numberOfBars="2">
        <bars>
                <color value="#33FF00"/>
                <color value="#CC0000"/>
        </bars>
</barchart>
```

O atributo *numberOfBars* especifica o número máximo de barras, sendo este um atributo opcional. No caso de este ser especificado, é necessário atribuir uma cor a cada uma das barras através do elemento *bars*.

O elemento *piechart* é semelhante ao elemento *barchart*, só que neste caso apenas se tem que especificar o raio do gráfico através do atributo *radius*.

```
<piechart numberOfSectors="2" radius="15">
</piechart>
```

Para definir estilos compostos utiliza-se o elemento compositeStyle. Este elemento não é nada mais do que uma lista de referências para estilos já definidos.

```
<compositeStyle>
    <styles>
        <style id="1"/>
        <style id="2"/>
    </styles>
</compositeStyle>
```

4.5.1.2 Elemento decisionTree

Na raiz do elemento *decisionTree* existem dois elementos: (*i*) *typeOfStyles*; (*ii*) *contexts*.

```
<decisionTree>
    <typeOfStyles>
            …
    </typeOfStyles>
    <contexts>
            …
    </contexts>
</decisionTree>
```

O elemento *typeOfStyles* consiste numa lista de elementos *typeOfStyle* que permite fazer uma descrição de tipos de estilos. O seu motivo será explicitado abaixo. O elemento *typeOfStyle* tem a seguinte estrutura:

```
<typeOfStyle id="1" description="Use visual property Brightness" >
    …
</typeOfStyle>
```

A descrição do tipo de estilo é realizada através dos elementos *simpleStyle* e/ou *compositeStyle*.

```
<simpleStyle>VariableBrightness</simpleStyle>

<compositeStyle>
    <simpleStyle>VariableBrightness</simpleStyle>
    <simpleStyle>VariableSize</simpleStyle>
</compositeStyle>
```

O elemento *contexts* consiste numa lista de elementos *context*. É o conjunto destes elementos que vai dar origem à árvore de decisão utilizada pelo componente *gestor de estilos*. A partir dos elementos *spatialObjects*, *numberOfNumericalColumns*, *numberOfAlphaNumericColumns* e *numberOfMeasures* é definido um contexto de uma possível interacção com o utilizador. É através do elemento *applicableStyles* que se descrevem os tipos de estilos aplicáveis.

```
<context id= "1">
    <spatialObjects>
        <spatialObject geometryType="point"/>
    </spatialObjects>

    <numberOfNumericalColumns>1</numberOfNumericalColumns>
    <numberOfAlphaNumericColumns>0</numberOfAlphaNumericColumns>
    <numberOfMeasures>1</numberOfMeasures>
    <applicableStyles>
        <typeOfStyle id="2"/>
    </applicableStyles>
</context>
```

Deste modo e com esta informação em memória, o gestor de estilos constrói uma tabela de *lookup*, onde a chave é *context* e o valor corresponde a uma lista de tipos de estilos aplicáveis. Com esta tabela o gestor de estilos é capaz de obter todas as definições dos estilos que sejam do tipo de estilos aplicáveis.

4.5.2 Elemento multidimensional

É no elemento *multidimensional* que é feita toda a descrição do modelo multi-dimensional (*star schema* ou *snowflake schema*). Na raiz do elemento *multidimensional* encontram-se os elementos: (*i*) *dimensions*, que contém vários elementos *dimension*; (*ii*) *cubes* que pode conter diversos elementos *cube*.

A estrutura base do elemento *dimension* é a seguinte:

```
<dimension>
        <levels/>
        <hierarchies/>
</dimension>
```

É através do elemento *dimension* que é feita a descrição das dimensões. Para cada dimensão é necessário descrever os diversos níveis e as suas hierarquias. O elemento *levels* contém a descrição dos diversos níveis e a referência para o nível base. Cada elemento *level* contém a descrição dos atributos, hierarquias e referência para os *levels* de granularidade superior. A sua estrutura é a seguinte:

```
<level id="14" primaryAttribute="38" displayAttribute="39" sortAttribute="39" spatialAttribute="40"
      tableRef="8" name="…">
    <attribute id="38" columnRef="71" name="…"/>
    <attribute id="39" columnRef="72" name="…"/>
    <attribute id="40" columnRef="73" name="…" spatial="true"/>
    <upperLevels>
        <upperLevel levelRef="15"/>
    </upperLevels>
    <hierarchies/>
    <preComputing>
        <distances tableRef="20">
            <from columnRef="502" />
            <to columnRef="503"/>
            <distanceValue columnRef="504"/>
        </distances>
    </preComputing>
</level>
```

Comparativamente com o meta-modelo anterior [**1**], foi introduzido o elemento *preComputing*. Este elemento está apenas presente quando se trata de um nível espacial. É nele que se descreve qual

a tabela que guarda o cálculo de distâncias entre os atributos espaciais. Este elemento pode vir a mostrar-se muito útil no processo de agrupamento espacial, como se verá no Capítulo 5.

O elemento *cube* tem a seguinte estrutura:

```
<cube id="" name="" factTableRef="" databaseRef="" mapserverRef="" description="">
    <maps/>
    <dimensions/>
    <measures/>
     <aggregates/>
    <aggregateChildren/>
</cube>
```

O elemento *maps* detém a configuração do mapa base, nível base de zoom e coordenadas centrais. Contém também a lista das *layers* e objectos espaciais utilizados neste cubo que foram descritas no elemento *mapservers*. O elemento *dimensions* é também uma lista de dimensões utilizadas neste cubo. No elemento *measures* encontra-se a definição das métricas numéricas. Para definir um agregado recorre-se ao elemento *aggregates* e o elemento *aggregateChildren* descreve uma hierarquia entre os agregados definidos.

4.6 Resumo

Nesta dissertação, a principal novidade na arquitectura do servidor consiste na incorporação de um gestor de tabelas que suportasse o desenvolvimento e integração das propostas realizadas nesta dissertação, no sistema SOLAP+ (dois atributos espaciais de diferentes dimensões e integração de agrupamento espacial).

Relativamente à arquitectura do cliente, a principal alteração consiste na introdução de um gestor de estilos que dá suporte a toda a gestão/construção de estilos. Associado ao gestor de estilos, no meta-modelo, foi concebida uma sintaxe para a definição de estilos e respectiva árvore de decisão.

Capítulo 5

Implementação

Este capítulo apresenta da perspectiva de implementação os novos componentes da arquitectura do sistema SOLAP+.

Este capítulo apresenta, de uma perspectiva de implementação, todos os componentes que foram introduzidos ao protótipo desenvolvido em [**1**]. Através desses componentes, novas interacções foram introduzidas ao protótipo.

O caso 6 foi incorporado (dois atributos espaciais de diferentes dimensões). Do ponto de vista de gestão de dados, tudo o que está relacionado com este caso de interacção é previsto. No entanto, devido a limitações do servidor de mapas, não foi possível criar estilos com gráficos associados aos arcos (necessário, por exemplo, quando se está na presença de atributos semânticos de dimensões semânticas). A interacção do utilizador proposta, no *caso 6* de interacção, não foi implementada.

Foi realizada a integração de algoritmos de agrupamento espacial ao protótipo. Esta integração prevê agrupamento de pontos, de polígonos e adapta-se a todos os casos de interacção implementados actualmente no protótipo, incluindo no caso 6 de interacção. Foi também implementado o algoritmo que restringe o agrupamento com base nas hierarquias espaciais. Em casos de interacção com agrupamento espacial, a implementação realizada prevê a utilização da tabela de detalhe para analisar os dados pertencentes a um grupo.

No protótipo foi também criado o *gestor de estilos*. Este gestor de estilos dá suporte à criação/alteração de estilos e permite uma legenda modificável. Do ponto de vista conceptual (ver Figura 66) foram implementados todos os componentes. Porém, o componente *interface* implementado é elementar e apenas permite a alteração de estilos.

5.1 Tecnologias

As tecnologias escolhidas por Ruben Jorge [**1**] para a implementação do protótipo foram as seguintes:

- **Metadados**: XML;
- **Servidor**: *java* e comunica com o cliente utilizando a tecnologia *Web Services*;
- **Cliente**: aplicação *web* desenvolvida com JSF e *Oracle maps*;
- **SGBD**: *Oracle*;
- **Servidor de Mapas**: *Mapviewer*.

No actual protótipo foram mantidas as tecnologias, migrando apenas da versão do *mapviewer* para a versão mais recente, pois permite a criação de estilos que não é possível nas versões anteriores.

5.2 Servidor

Como foi referido no capítulo anterior, foi introduzido o *gestor de tabelas* à arquitectura do servidor. Este é constituído pelos seguintes componentes: (*i*) construtor; (*ii*) módulo de agrupamento; (*iii*) geradores de novos objectos espaciais (ex: gerador de arcos).

O *construtor* é responsável por criar e preencher a nova tabela com base no conjunto de dados e nas novas representações espaciais.

5.2.1 Módulo de Agrupamento

Quando um conjunto de dados entra na fase de pré-processamento, este é de imediato sujeito à primeira fase que avalia se realmente é necessário aplicar o restante processo de agrupamento. Nesta fase não foi implementada qualquer heurística, mas ainda assim, a arquitectura da solução já prevê a integração de heurísticas.

É no módulo de agrupamento que se aplicam os algoritmos de agrupamento espacial sobre os objectos espaciais. O algoritmo implementado para o agrupamento de pontos foi o DBSCAN [**21**] e para o agrupamento de polígonos foi o P-DBSCAN [**32**].

No caso de agrupamento de polígonos o cálculo da distância de *hausdorff* entre polígonos requer algum processamento. Esse processamento impediria dar uma resposta eficiente quando o utilizador pretendesse ter um caso de interacção com agrupamento de polígonos. Com o objectivo de contornar esta questão, as distâncias entre polígonos são pré-calculadas e guardadas numa tabela. Daí o elemento *preComputing* (no meta-modelo) ter um papel importante no agrupamento espacial de polígonos. Para o cálculo da distância de *hausdorff* original foi implementado o algoritmo apresentado em [**37**].

Outra questão pertinente consiste no parâmetro de entrada *eps* que ambos os algoritmos têm. Com o objectivo de dar resposta a este problema, os autores do algoritmo DBSCAN [**21**] propõem uma heurística. Em [**21**] são calculadas as distâncias de cada objecto espacial ao seu *k* vizinho mais próximo. Estas distâncias são posteriormente ordenadas e colocadas num gráfico a que designam de *sorted k-dist graph* (Figura 73 retirado de [**31**]). Este gráfico dá algumas pistas da distribuição da densidade dos objectos espaciais. Ao escolher um ponto *p* e ao atribuir o parâmetro *eps* ao valor *k-dist(p)*, todos os pontos à direita serão considerados *core objects*, enquanto que os pontos à esquerda não serão agrupados. Deste modo, os autores propunham que o utilizador escolhesse o ponto que se

encontrava no primeiro "vale" no gráfico, pois sustentam que o valor da função *k-dist* desse ponto corresponde à máxima distância do menor grupo.

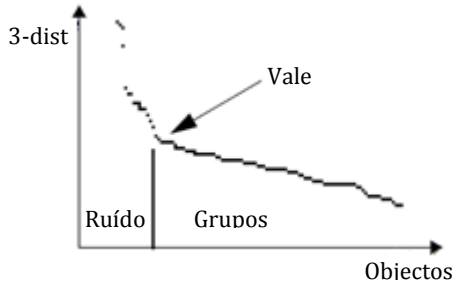


Figura 73 - *Sorted 3-dist graph*

No entanto, esta heurística requer interacção com o utilizador e, no modelo genérico SOLAP, esse requisito torna a interacção pouco atractiva. Deste modo, para tornar o processo mais automatizado foi desenvolvida uma nova heurística. Esta heurística não tem o objectivo de descobrir apenas um valor adequado, mas sim obter mais do que um valor de modo a permitir ao utilizador escolher entre um agrupamento com menos ou com mais grupos de forma intuitiva. De notar que esta heurística adapta-se a qualquer função de distância entre os objectos espaciais.

Assim, propomos uma heurística que procura por "quebras" na função *k-dist*. Nas quebras é onde se verifica o ponto de viragem para que níveis de densidades se vão agrupar pontos. Por exemplo, na Figura 74, ao considerar a quebra *Q3* apenas se iriam considerar grupos onde se verificassem zonas muito densas de objectos espaciais.

As quebras são obtidas da seguinte forma: os valores da função *k-dist* são percorridos por ordem crescente e, por cada ponto, é calculado o valor *k-dist*(*p2*)- *k-dist*(*p1*) e é actualizada a média destes valores; quando *k-dist*(*p2*)- *k-dist*(*p1*) é maior que *factor*media* é considerada uma quebra; nesse caso, é inicializado o valor da média a zero e repete-se o mesmo procedimento até percorrer todos os valores da função. Caso não se obtenha qualquer quebra com o *factor* utilizado, é repetida a heurística, mas com um valor menor. Na nossa implementação o valor de *factor* é inicializado a três.

A complexidade da heurística apresentada é $O(n)$ em que n representa o número de objectos envolvidos no processo de agrupamento.

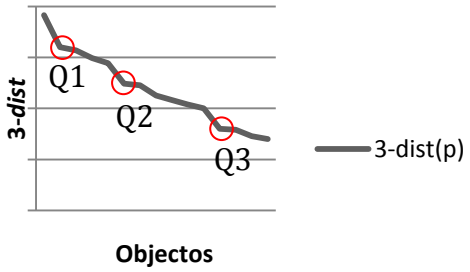


Figura 74 - Exemplo de um possível *sorted dist graph* e as suas quebras.

O parâmetro de *k* segue a heurística proposta em [**31**] que propõe a seguinte fórmula: *k* = 2*dimensão-1, onde dimensão se refere à dimensionalidade dos dados.

Durante a execução dos algoritmos é construída uma tabela que faz a correspondência entre o atributo semântico de um objecto espacial e o grupo a que este pertence (no caso de pertencer a algum). Com esta informação, o *rowset* é processado e este é devolvido ao *construtor* já com os dados agregados por grupo.

Considere que a *fábrica 1* pertence ao mesmo grupo que a *fábrica 2*. O atributo *meio* é um atributo semântico de uma dimensão semântica. Eis o resultado do processamento do *rowset*:

	Meio	
	Ar	Agua
Indústria	**Soma Qtd Emitida**	**Soma Qtd Emitida**
Fábrica1	50	60
Fábrica 2	20	30

→

	Meio	
	Ar	Agua
Indústria	**Soma Qtd Emitida**	**Soma Qtd Emitida**
Group 0	70	90

Figura 75 - Ilustra o processamento do *rowset* sujeito a agrupamento.

O processamento do *rowset* é realizado em apenas uma iteração, com vista a introduzir o menor tempo possível no tempo de resposta. Duas linhas são agregadas quando estas "pertencem" ao mesmo grupo e detêm os mesmos valores dos atributos semânticos.

5.2.2 Geradores de Novos Objectos Espaciais

Associado ao gestor de tabelas foram construídos diversos geradores de novos objectos espaciais. São eles: (*i*) gerador de arcos; (*ii*) gerador de representações para os grupos de objectos espaciais.

O gerador de arcos é responsável por criar os arcos no *caso 6* de interacção. O gerador de arcos implementado constrói o arco apenas com base nas extremidades. Não tem qualquer mecanismo inteligente que, com base num conjunto de arcos, os "arrume" de modo a que estes fiquem mais legíveis.

O gerador de representações de grupos de objectos é responsável por gerar os objectos espaciais dos grupos de objectos. Para os pontos, a representação implementada foi o centróide do grupo e no caso dos polígonos, a representação implementada foi a união destes. Uma outra representação pode ser aplicável.

5.3 Cliente

Na arquitectura do cliente a principal alteração foi a introdução do componente *gestor de estilos*. Ainda assim, a *interface* também foi alvo de pequenas alterações.

5.3.1 Gestor de Estilos

O principal objectivo do gestor de estilos é converter uma definição concreta de um estilo num mapa temático, como pode ser observado na Figura 76, em que é aplicado o estilo composto com identificador 9 a uma dada interrogação.

```
    <style id="7">
        <variableSize marker="star" initialSize="10"
increment="6" numberOfClasses="5"
typeOfDistribution="NATURALBREAKS">
        <sizeDefinition/>
        <intervals/>
        </variableSize>
    </style>
    <style id="8">
    <variableBrightness baseColor="cyan" numberOfClasses="5"
typeOfDistribution="NATURALBREAKS">
        <gradientDefinition/>
        <intervals/>
        </variableBrightness>
    </style>
    <style id="9">
        <compositeStyle>
            <styles>
                <style id="7"/>~
                <style id="8"/>
            </styles>
        </compositeStyle>~
    </style>
```

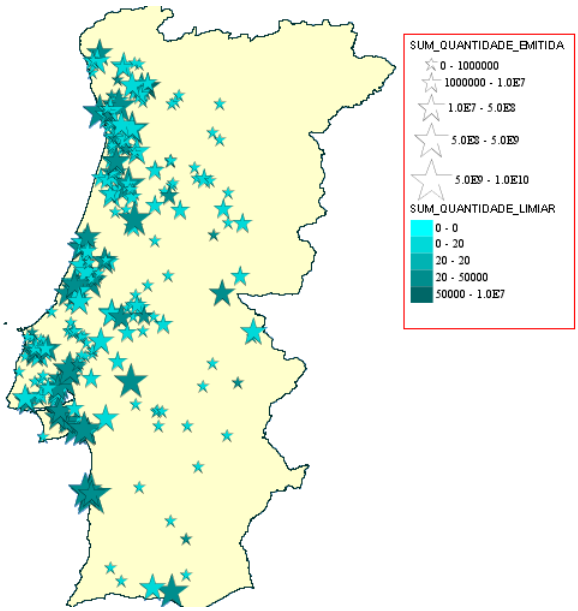


Figura 76 - Definição de estilo e o mapa exemplificativo da sua aplicação.

Em redor do gestor deste componente encontram-se as seguintes etapas:

1. Escolha do tipo de estilo mais apropriado;
2. Escolha da instância de estilo mais apropriada;
3. Construção da legenda do estilo;
4. Gerar o código necessário para se realizar o pedido de mapa ao servidor de mapas (em concreto gerar o pedido XML para ser enviado ao *mapviewer*).

Na implementação realizada foram abordadas todas as tarefas, à excepção das primeiras duas etapas. Em ambas as etapas, quando existe mais do que uma possibilidade, a abordagem seguida é a escolha da primeira.

Relativamente ao processo de construção da legenda, são suportadas duas formas de distribuição de dados: (*i*) uniforme; (*ii*) *natural breaks*[4] [**38**]. Sobre os valores que dividem os dados em

[4] É um processo de classificação de dados que determina o melhor arranjo dos dados em diferentes classes. O resultado é obtido reduzindo a variância dentro das classes e maximiza a variância entre as classes.

diferentes classes é aplicado o algoritmo [**39**]. Com este algoritmo, em vez dos valores das legendas serem pouco expressivos e potencialmente com casas decimais, tornam-se em apenas em números sem casas decimais, o que facilita a compreensão da legenda (ver Figura 76).

5.3.2 Interface

À interface original do protótipo (ver Figura 12, na página 31) foi introduzido um painel lateral designado de *Clustering Control*.

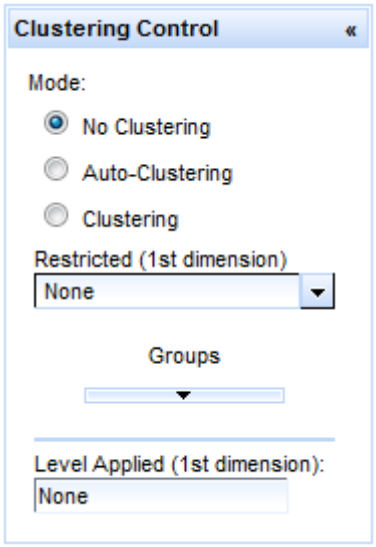


Figura 77 - Painel *Clustering Control.*

Através deste painel, o utilizador escolhe o modo de agrupamento e na *combo box* escolhe a hierarquia espacial. Se não escolher nenhuma hierarquia aplica o agrupamento *adhoc*. Ao escolher uma hierarquia, no campo abaixo, é disponibilizado o nível da hierarquia pelo qual se está a restringir o agrupamento. É no *slider* onde tem a possibilidade, de forma intuitiva, de optar por obter menos ou mais grupos. Este contém cinco valores possíveis (-2, -1, 0, 1, 2). Internamente, este *slider* "navega" sobre as quebras retornadas pela heurística. No entanto, o utilizador está completamente abstraído dela. Para ele o valor zero corresponde a um agrupamento normal e o valor negativo ou positivo corresponde a um menor ou maior número de grupos, respectivamente.

Capítulo 6

Caso de Estudo e Validação

Este capítulo apresenta o caso de estudo e exemplos de interacção com o objectivo de validar as propostas realizadas nesta dissertação.

Foram implementadas as propostas realizadas nesta dissertação no protótipo actual. Com o objectivo de validar uma parte dessas propostas (a interacção com dois atributos espaciais de diferentes dimensões) foi criada uma base de dados adequada, a partir de dados fornecidos pela *ViaTecla*[5], e o respectivo meta-modelo XML. Assim, numa primeira parte, são testados e validados os casos de interacção com dois atributos espaciais de diferentes dimensões através do conjunto de dados anterior. Numa segunda fase, será testada e validada a integração do agrupamento espacial com um conjunto de dados referente a emissões de poluentes, herdado da dissertação [**1**].

6.1 Caso de Estudo 1: Viagens e Turismo

Viagens e Turismo é um dos sectores económicos mais importantes e com rápido crescimento, produzindo empregos e riqueza substancial para as economias nacionais em todo o mundo. Desde 1990, a organização WTTC (*World Travel & Tourism Council*) compromete-se a oferecer uma pesquisa anual sobre o impacto deste sector nas economias nacionais do globo. Outras organizações e as próprias empresas procuram indicadores dos mercados ao nível nacional, através da análise estratégica dos factores chave que influenciam o mercado, tais como, rendimento disponível, férias, feriados e hábitos dos clientes, etc. A análise de conjuntos de dados espaciais com dados relativos a reservas de viagens pode ajudar a: (*i*) obter indicadores da "atractividade" dos países, através da observação do número total de passageiros que saem e entram num país [**40**]; (*ii*) identificar destinos com elevada ou baixa procura; (*iii*) relacionar a procura dos destinos com as épocas; (*iv*) relacionar destinos com a informação de *background* dos clientes (salário, educação, idade, etc.); (*v*) outras análises.

O primeiro caso de estudo baseia-se num conjunto de dados de reservas de voos. Uma reserva de voo é constituída pelo aeroporto de partida e pelo aeroporto destino. A esta informação está associado o tipo de reserva (apenas ida, ida e volta, etc). O conjunto de dados está modelado da seguinte forma:

[5] A VIATECLA é uma *software house* especializada na construção de soluções e produtos, críticos ao negócio dos seus clientes. A sua estratégia de actuação tem-se centrado no desenvolvimento de produtos próprios e em serviços de elevado valor acrescentado. No contexto deste trabalho realça-se a plataforma *VIATECLA KEYforTravel* responsável pela agilidade de operação das empresas do sector do turismo, abrangendo áreas tão diversas como a da definição de produto, contratação, promoção e suporte ao processo de venda.

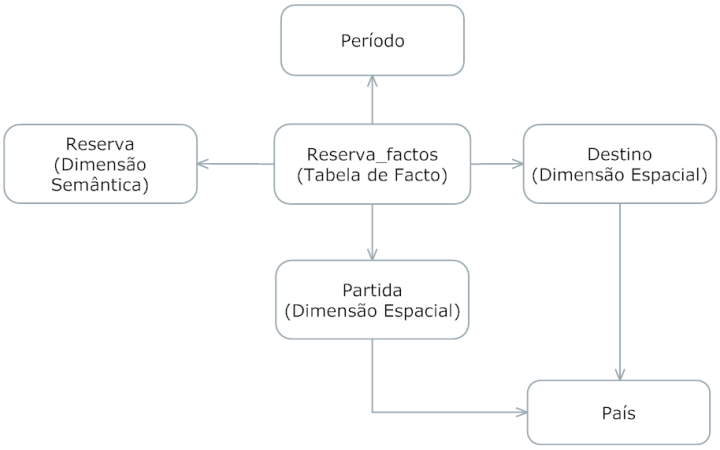


Figura 78 - Modelo de dados utilizado no caso de estudo.

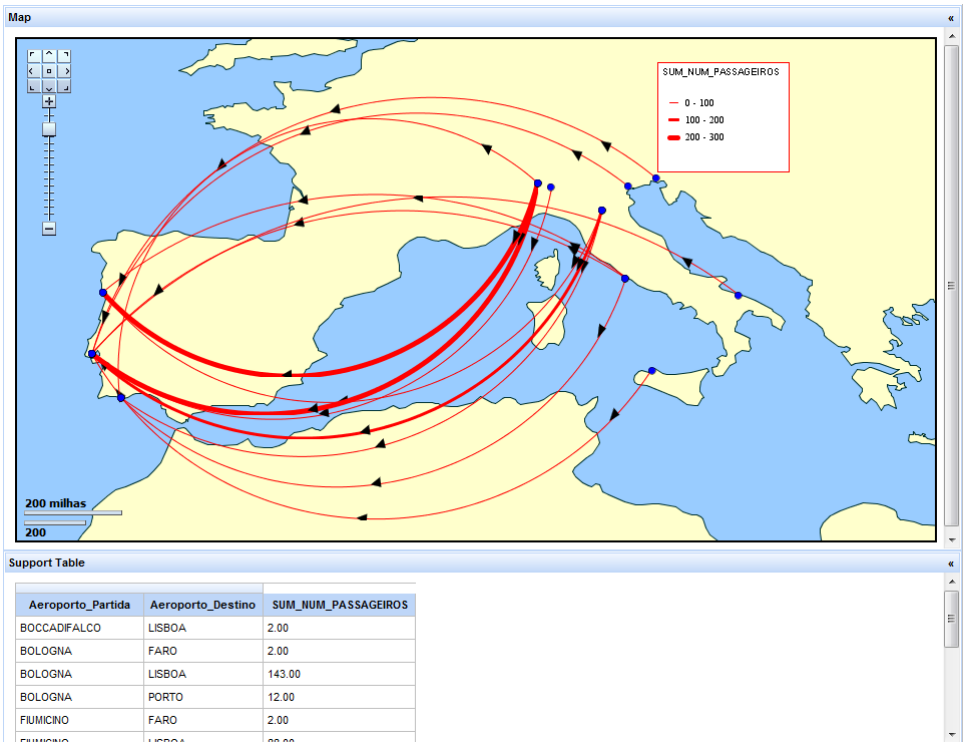
As dimensões *Partida* e *Destino* são dimensões espaciais. Estas contêm dois atributos espaciais: a localização do aeroporto e o país, os quais formam a seguinte hierarquia espacial: **aeroporto - país**. A tabela de facto contém as seguintes métricas: (*i*) número de passageiros; (*ii*) taxas; (*iii*) preço de custo.

Uma série de exemplos de interacção, utilizando o protótipo com o conjunto de dados descrito anteriormente, são apresentados. Esses exemplos focam-se sobretudo nas propostas que foram realizadas e implementadas nesta dissertação.

6.1.1 Partida em Itália e Destinos para Portugal

Neste primeiro exemplo, o objectivo é analisar o total de passageiros que têm reservas com partida nos aeroportos de Itália e com destino nos aeroportos de Portugal. Para tal, são utilizados os níveis desejados a que se quer observar os dados (*aeroporto* em ambas as dimensões), assim como a métrica numérica (soma do número de passageiros). Foi também realizado o *slice* sobre o atributo *país* em cada uma das dimensões espaciais.

O estilo utilizado definido no meta-modelo para *contextos* com **dois objectos espaciais** e uma **coluna numérica** foi o *Estilo Simples*(*Tamanho*) associado a uma distribuição uniforme com três classes.



Aeroporto_Partida	Aeroporto_Destino	SUM_NUM_PASSAGEIROS
BOCCADIFALCO	LISBOA	2.00
BOLOGNA	FARO	2.00
BOLOGNA	LISBOA	143.00
BOLOGNA	PORTO	12.00
FIUMICINO	FARO	2.00

Figura 79 - Mapa e tabela de suporte (Partidas de Itália com Chegadas a Portugal).

A relação de 1:1 entre a tabela de suporte e o mapa é verificada (Figura 79). Através da visualização do mapa (ver Figura 79) é observável que no norte de Itália existe um maior número de reservas para Portugal. Podemos também constatar que a escolha com destino Lisboa é semelhante à cidade do Porto, e com valores relativamente elevados.

6.1.2 Partidas em Portugal e Espanha com destino nos seus Arquipélagos

O segundo exemplo tem como objectivo analisar a procura pelas ilhas de Portugal e Espanha, com origem nos mesmos. Com este objectivo, queremos visualizar as partidas ao nível de país, mas os destinos queremos observá-los ao nível de aeroporto. Mantém-se a métrica utilizada anteriormente, assim como o estilo (à excepção da distribuição dos dados, onde é utilizada a partição *natural breaks* com quatro classes). Foram também realizados os devidos *slices* para apenas seleccionar os dados com partidas em Portugal e Espanha e destino os aeroportos das respectivas ilhas.

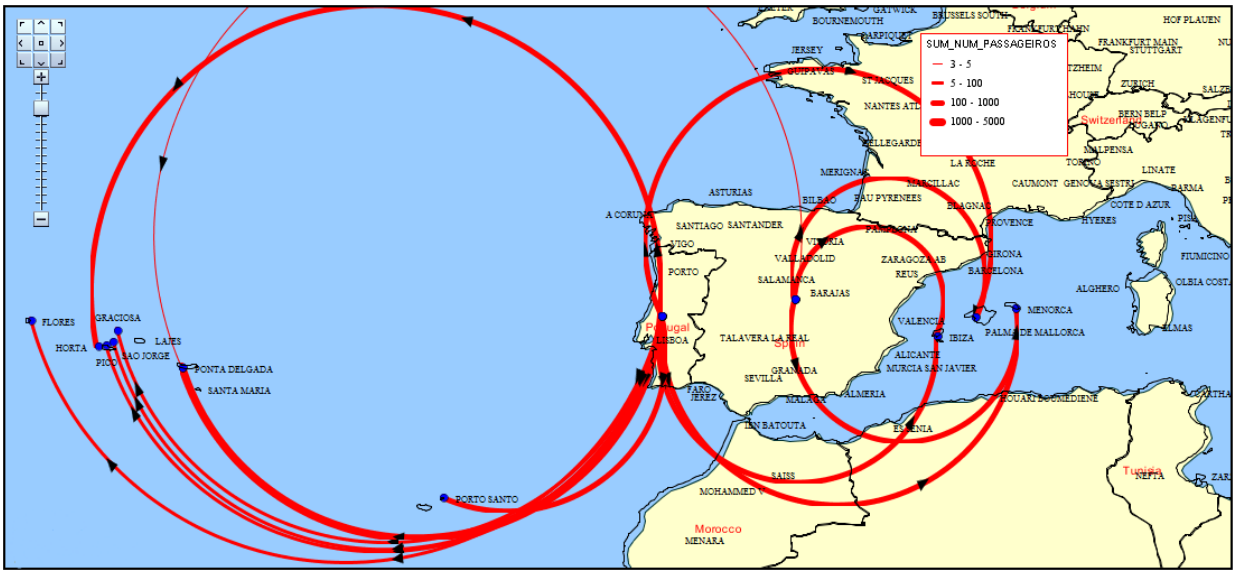


Figura 80 - Mapa das Partidas de Portugal e Espanha c/ destino os seus arquipélagos.

Mais uma vez a relação de 1:1 mantém-se e, visualmente, retira-se quais os destinos com maior procura de ambos os países.

6.1.3 Partidas de Portugal para Brasil

Com este exemplo vamos assumir que queremos visualizar todas as reservas que têm como partida aeroportos de Portugal e destino aeroportos do Brasil. São utilizados os níveis aeroportos das dimensões espaciais partidas e chegadas. O estilo definido no meta-modelo é semelhante ao anterior, mas com apenas três classes. O resultado obtido foi o seguinte (Figura 81):

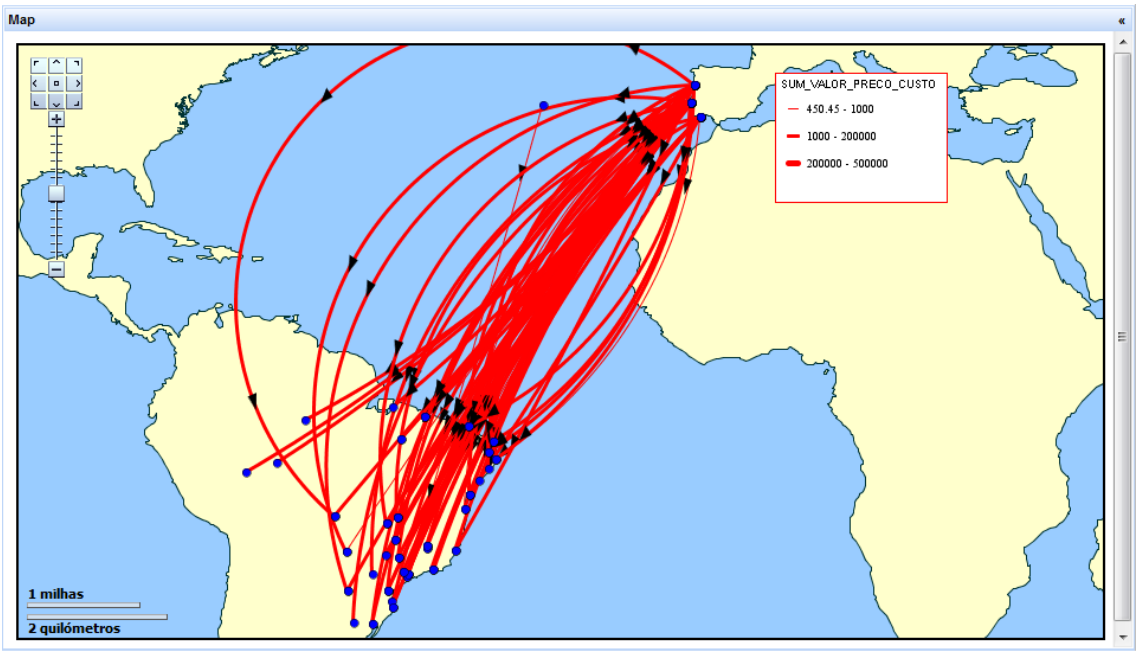


Figura 81 - Exemplo 4: mapa (interacção sem agrupamento espacial).

Esta interacção provoca um grande número de resultados, e, como consequência muita desorganização no mapa. Relembro que o número de resultados adequados depende daquilo que visualmente é possível analisar. Com o objectivo de sumarizar ainda mais os resultados, é utilizado o agrupamento *adhoc*, obtendo o resultado da Figura 82.

Uma vez que se está numa interacção com agrupamento, pode-se utilizar a tabela de detalhe para analisar em mais detalhe as "relações" que envolvam grupos. Neste caso em concreto, poder-se-ia procurar pelos aeroportos que contribuíam mais para o total da relação *Faro-Group0*.

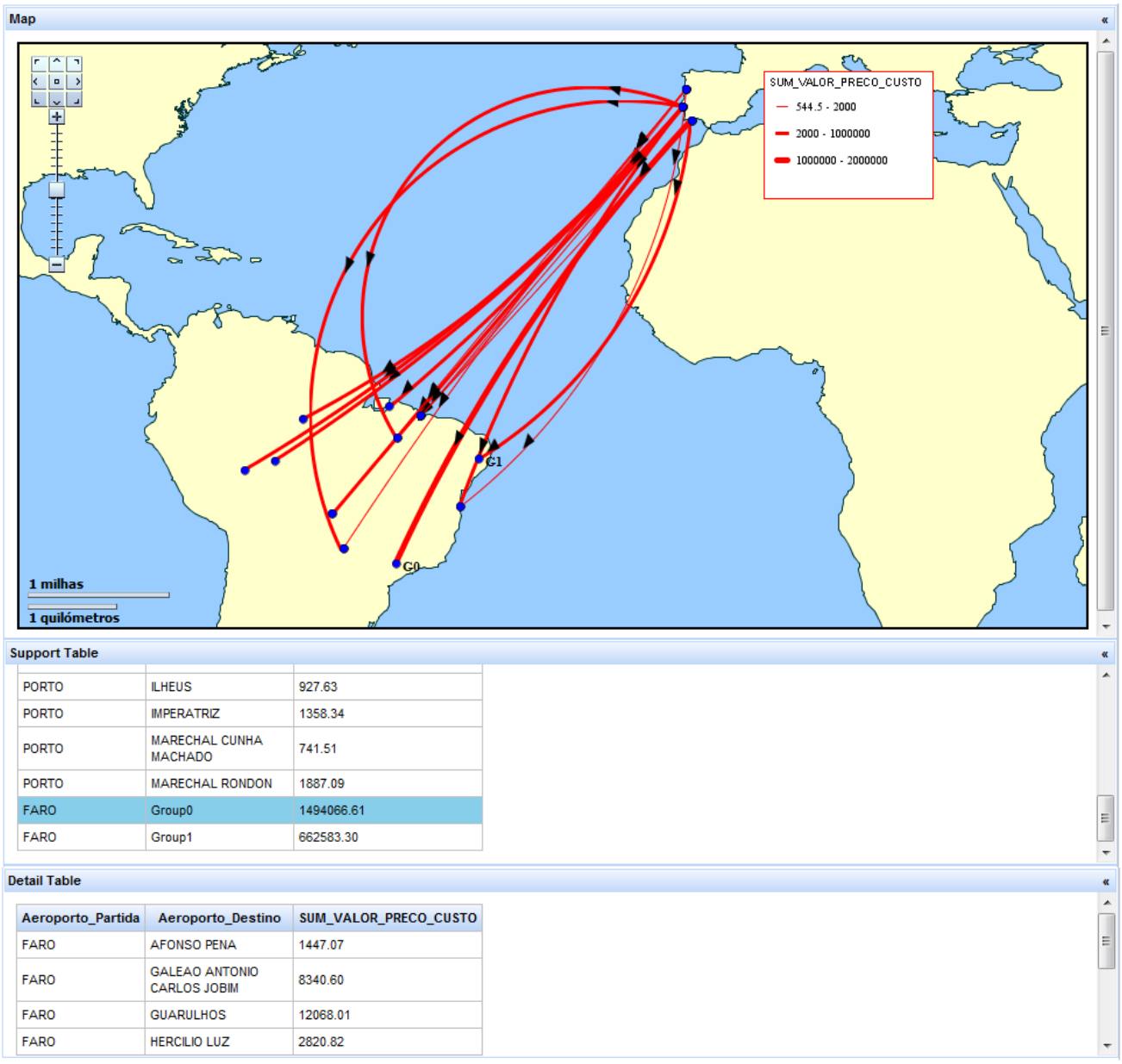


Figura 82 - Exemplo 4: mapa, tabela suporte e detalhe (c/ agrupamento espacial).

Posteriormente poder-se-ia utilizar o painel *clustering control* para obter mais ou menos grupos e ainda restringir os grupos pelas hierarquias espaciais. Estas questões serão focadas na secção 6.2.

6.2 Caso de Estudo 2: Emissões de Poluentes

Ao contrário do caso de estudo anterior, neste caso de estudo o objectivo é testar e validar o impacto que a integração de agrupamento espacial, no protótipo desenvolvido, tem na componente mapa.

O caso de estudo baseia-se no conjunto de dados sobre emissões de poluentes em Portugal. As informações relevantes sobre o modelo de dados para o caso de estudo são as seguintes:

- O modelo contém uma dimensão espacial designada de *Instalação*. Esta dimensão contém cinco atributos espaciais: localização, bacia hidrográfica, freguesia, concelho, distrito.
- A dimensão *Instalação* contém o atributo semântico *sector*, que designa o sector da actividade da indústria.
- Existem duas hierarquias espaciais na dimensão *Instalação:*
    1. Localização – Bacia Hidrográfica;
    2. Localização – Freguesia – Concelho – Distrito.

Os exemplos que se seguem estão apenas associados a um poluente. Quando se utiliza o nível *localização* com uma métrica (neste caso a soma da quantidade emitida) o mapa resultante sem qualquer agrupamento é o seguinte:

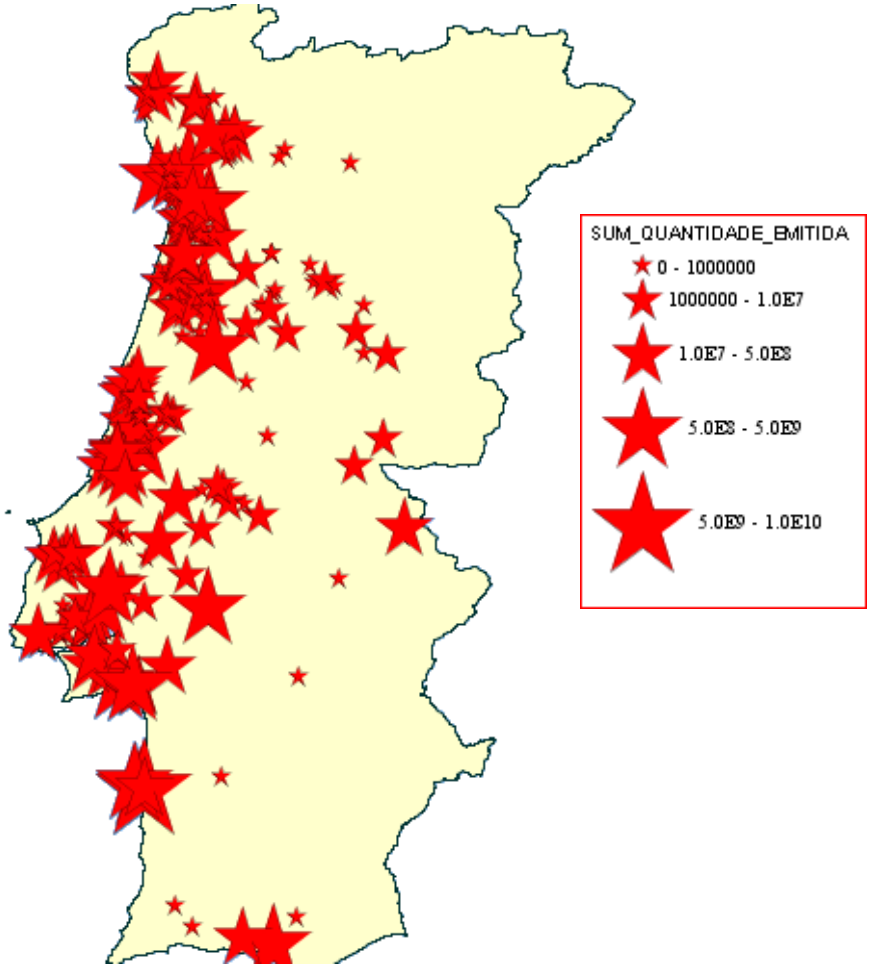


Figura 83 - Mapa base do segundo caso de estudo.

6.2.1 Heurística

Neste exemplo são mantidos os elementos de análise anteriores (atributo espacial e métrica), mas neste caso com a aplicação de agrupamento *adhoc*. Na figura seguinte, são disponibilizados os mapas para cada valor do *slider groups* (do valor -2 para o 2).

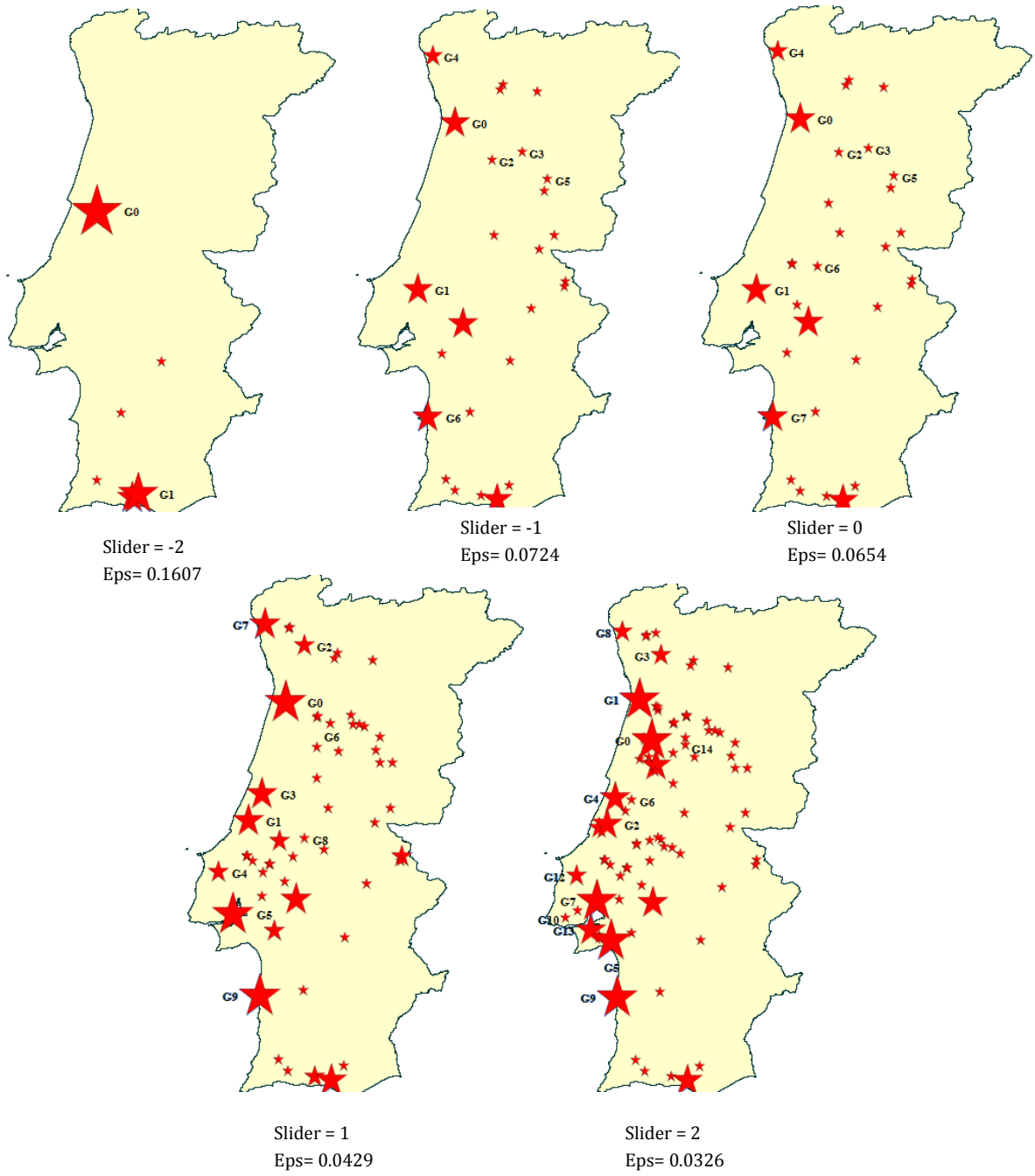


Figura 84 - Mapa para cada valor da posição do *slider groups*.

À medida que se avança na posição do *slider*, verifica-se um aumento do número de grupos. Através da heurística implementada, o utilizador está abstraído dos parâmetros de entrada dos algoritmos. O grupo é representado pelo centróide associado com o identificador *G*. Outra representação dos grupos interessante seria o menor polígono formado pelo grupo, e, neste caso em concreto, permitiria ter uma noção da área formada pelos grupos de instalações que emitem um determinado poluente.

6.2.2 Utilização de uma Hierarquia Espacial

Este exemplo mantém a interacção inicial do mapa da alínea *c* da Figura 84, só que desta vez o agrupamento é restringido pela segunda hierarquia espacial. Após obter os resultados, foi colocada como informação contextual a *layer* dos distritos. Eis o mapa resultante e respectivo painel *clustering control*:

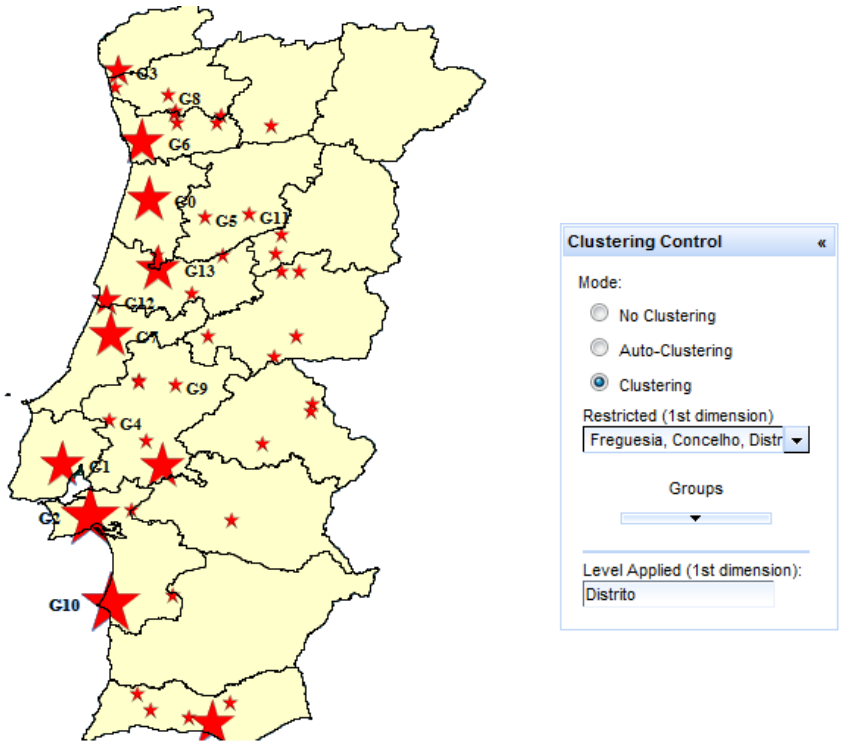


Figura 85 - Mapa com agrupamento restringido ao nível distrito.

Com o aumento do nível de zoom do mapa, o nível da hierarquia com que o algoritmo restringe o agrupamento não é o nível *distrito*, mas sim o nível *concelho*. Agora foi colocada a *layer* dos concelhos como informação contextual. O mapa resultante é o seguinte:

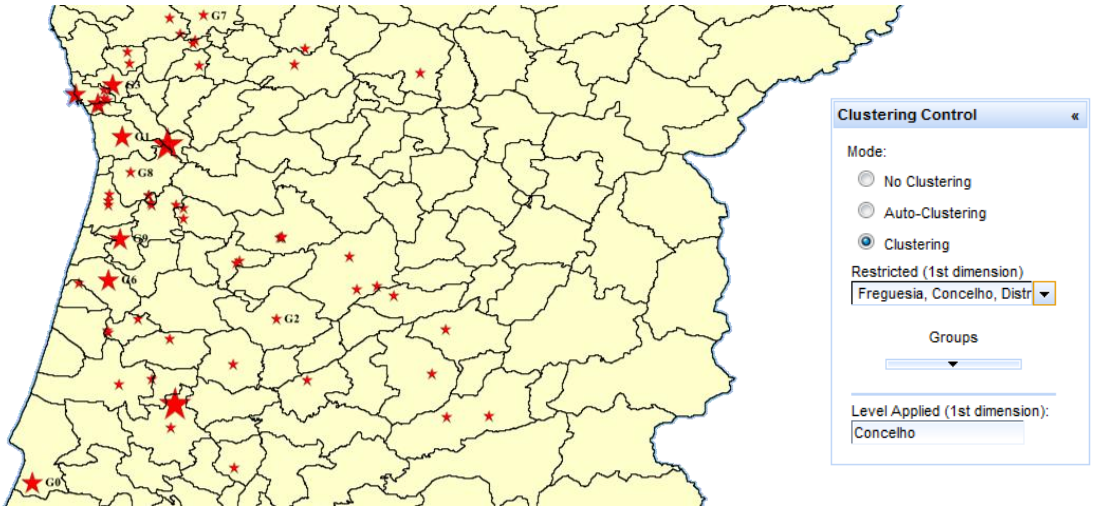


Figura 86 - Mapa com agrupamento restringido ao nível concelho.

6.2.3 Atributo semântico da dimensão espacial a um nível superior

Nos exemplos de agrupamento anteriores, apesar de se apenas disponibilizar o mapa, não têm estado presentes atributos semânticos da dimensão espacial a um nível de granularidade superior, comparativamente com o atributo espacial. Assim, com base na interacção inicial do mapa da alínea *c* da Figura 84, foi realizado um *slice* sobre o atributo *sector* e foram escolhidos cinco sectores. Este atributo foi posteriormente adicionado à tabela de suporte.

O estilo definido no meta-modelo para *contextos* com **um objecto espacial, uma coluna alfanumérica** e **uma coluna numérica** foi o *Estilo Composto(Estilo Simples(Tamanho), EstiloSimples(Cor))*.

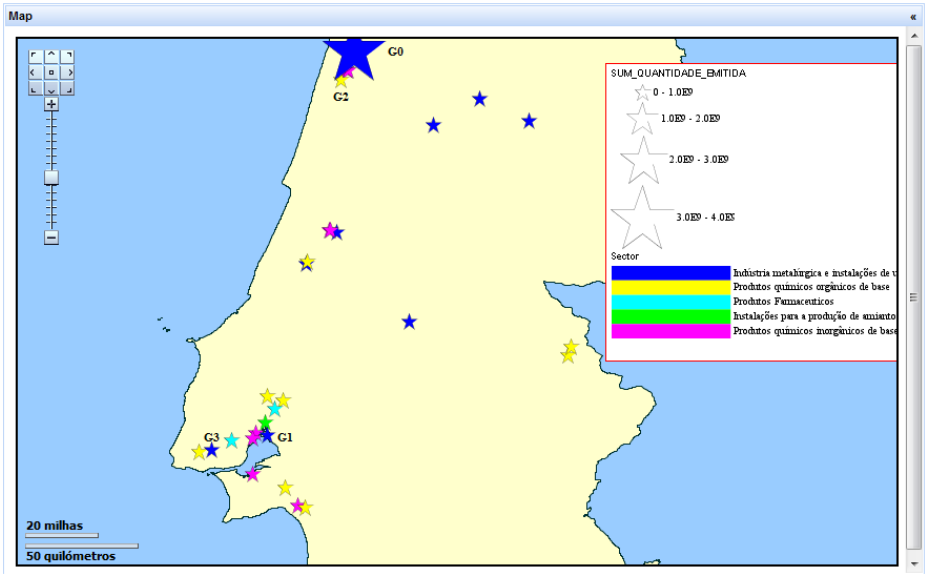
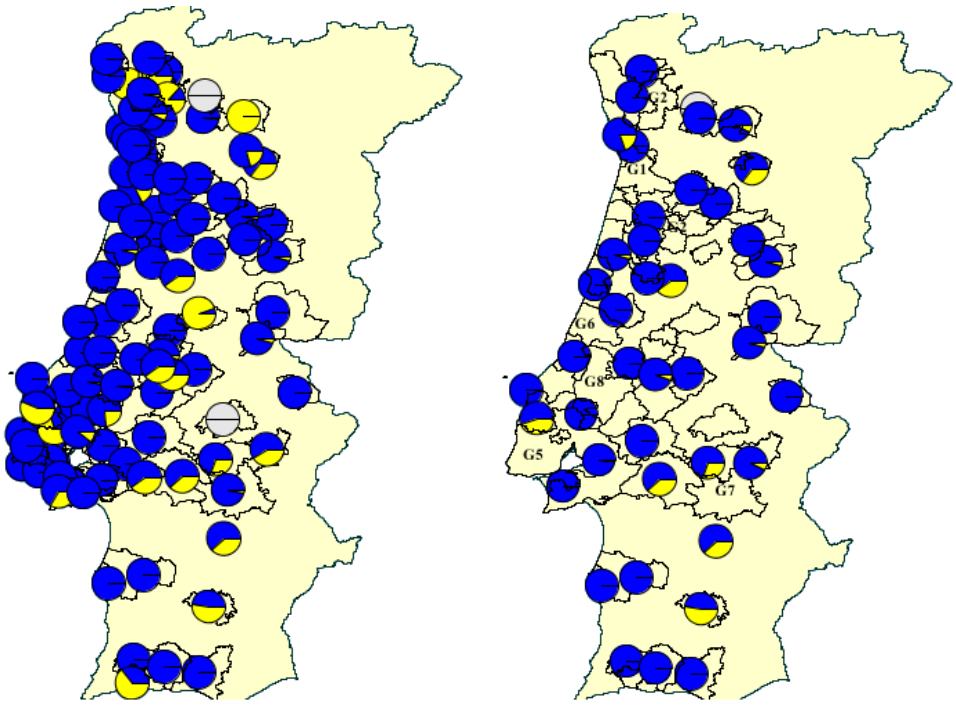


Figura 87 - Exemplo 8: Mapa.

Tal como foi proposto nestes casos de interacção, não deve haver perca das propriedades da análise e, assim, apenas foram agrupadas as indústrias que partilhem o mesmo sector de actividade. Este tipo de interacção torna-se útil para descobrir *hotspots* de indústrias com o mesmo tipo de actividade.

6.2.4 Polígonos

Assumindo que, em vez de se querer visualizar os dados ao nível de instalação, quer-se visualizar os dados ao nível de concelho. Agora já não se está perante a forma geométrica *ponto*, mas sim perante polígonos. Ao ser utilizado um dado atributo semântico que gera duas colunas numéricas, o resultado obtido é o seguinte:



a) b)

Figura 88 - Exemplo 8: a) sem agrupamento; b) com agrupamento

Após a aplicação do agrupamento espacial *adhoc*, reduziu-se consideravelmente o número de gráficos associados ao mapa, permitindo uma melhor leitura dos gráficos.

Capítulo 7

Conclusão e Trabalho Futuro

Este capítulo apresenta as conclusões obtidas nesta dissertação e dá direcções para trabalho futuro.

Este capítulo apresenta as conclusões finais do contributo que esta dissertação realizou na área SOLAP e dá direcções para trabalho futuro.

7.1 Conclusão

O propósito principal desta dissertação era, por um lado, desenvolver um modelo de interacção que suportasse análises com duas dimensões espaciais e, por outro, que integrasse a utilização de algoritmos de agrupamento espacial nos sistemas SOLAP. Uma solução foi definida para cada objectivo e uma extensão foi proposta ao modelo genérico SOLAP [**1**]. Além dos objectivos anteriores, foi proposto um *gestor de estilos*. Este tem incorporado um modelo de estilos, também ele definido nesta dissertação, que visa facilitar ao utilizador a interpretação de mapas temáticos.

Após a concepção da extensão do modelo genérico SOLAP [**1**], foram implementadas no protótipo existente a maior parte das propostas realizadas. Neste processo, foram definidos novos componentes e realizados acertos na arquitectura do sistema. O protocolo de comunicação e o meta-modelo foram também alvo de pequenas alterações e de novos elementos.

Uma nova heurística (sem interacção com o utilizador) foi concebida, que determina possíveis valores para o parâmetro *eps* dos algoritmos DBSCAN e P-DBSCAN.

Com as propostas implementadas no protótipo, este foi utilizado em dois casos de estudo que exemplificam e validam os objectivos desta dissertação.

7.2 Trabalho Futuro

As direcções futuras podem ser vistas sobre duas perspectivas. Primeiro, o trabalho realizado necessita ainda de futuras contribuições. Cito as mais relevantes:

- Para o *caso 6* de interacção, desenvolver um mecanismo inteligente que não se baseie apenas nas extremidades dos arcos. Eventualmente esse mecanismo poderá utilizar outras formas "arbitrárias" em vez de arcos e atribuir as diferentes formas consoante a distância a que estão as extremidades;
- Relativamente ao agrupamento espacial, desenvolver e suportar outras formas de representação de grupos. Integrar essas formas de representação com o processo de construção de mapas temáticos, tendo em conta os possíveis conflitos que podem surgir;
- Ainda relacionado com agrupamento espacial, implementar uma fase de avaliação da necessidade de agrupamento. Relacionado com a medida de visualização, é necessário

estudar as medidas de dispersão espaciais existentes e potencialmente desenvolver uma nova medida;

- No *gestor de estilos*, o processo de escolha do estilo apropriado para a representação do mapa é complexo. Apesar de nesta dissertação se verificar algum avanço neste processo, muitas questões não foram abordadas.

Doutra perspectiva existem muitas questões em aberto na actualidade dos sistemas SOLAP. Se nesta dissertação foi realizada a integração de agrupamento espacial, outras formas de agrupamento podem proporcionar análises interessantes, particularmente:

- Agrupamento com base em atributos não espaciais;
- Agrupamento espácio-temporal;
- Combinar agrupamento espacial com agrupamento com base em atributos não espaciais.

Actualmente a integração das métricas espaciais é um assunto que ainda necessita de uma investigação profunda de modo a se obter um modelo bem definido e utilizável.

Finalmente, é necessário facilitar o fluxo desde as fontes de dados até aos cubos espaciais, como por exemplo, produzir os dados espaciais a diferentes níveis de granularidade a partir das coordenadas longitude e latitude.

Bibliografia

[1] Ruben Jorge, "SOLAP+:Extending the Interaction Model," Universidade Nova de Lisboa - Faculdade Ciências e Tecnologia, Monte de Caparica, Tese de Mestrado 2009.

[2] G. Kramer, "The Practical Union of OLAP Analysis and Geographic Mapping," *ArcUser Online*, 2006.

[3] S. Rivest, M. J. Proulx Bédard Y., "Spatial On-Line Analytical Processing (SOLAP): Concepts, Architectures, and Solutions from a Geomatics Engineering Perspective," in *Data Warehouses and OLAP: Concepts, Architectures and Solutions*.: IRM Press, 2007, pp. 298-319.

[4] Sonia Rivest and Yvan Bédard and Pierre March, "Towards better support for spatial decision-making: defining the characteristics," in *Geomatica*., 2001, pp. 539--555.

[5] Ralph Kimball Margy Ross, *The Data Warehouse Toolkit*, 2nd ed., 2002.

[6] Inmon, *Building the Data Warehouse*, 3rd ed., 2002.

[7] Matthew and Parmanto, Bambang Scotch, "SOVAT: Spatial OLAP Visualization and Analysis Tool," *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*, vol. 6, p. 142.2, 2005.

[8] KHEOPS. (2010) JMAP Spatial OLAP. [Online]. http://www.kheops-tech.com/en/jmap/solap.jsp

[9] Pascal Wehrle, Anne Tchounikine, and Maryvonne Miquel Sandro Bimonte, "GeWOlap: A Web Based Spatial OLAP Proposal," in *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Academy Doctoral Consortium, RDDS, SWWS, and SeB*, Robert Meersman and Zahir Tari and Pilar Herrero, Ed.: Springer, 2006, pp. 1596-1605.

[10] R Matias, "Integração de Informação Geográfica em Sistemas OLAP," 2006.

[11] M. Vitorino and R Caldeira, "The Spatial One," Universidade Nova de Lisboa - Faculdade de Ciências e Tecnologias, 2008.

[12] Bambang Parmanto and Valerie Monaco Matthew Scotch, "Usability Evaluation of the Spatial OLAP Visualization and Analysis Tool (SOVAT)," *Journal of Usability Studies*, vol. 2, pp. 76-95, Fevereiro 2007.

[13] Sandro Bimonte and Anne Tchounikine Maryvonne, "Spatial OLAP: Open Issues and a Web Based Prototype," in *10th AGILE International Conference on Geographic Information Science*, Aalborg University, Denmark, 2007, pp. 1-11.

[14] Osmar R. Zaïane and Andrew Foss and Chi-hoon Lee and Weinan Wang, "On Data Clustering Analysis: Scalability, Constraints and Validation," in *Advances in Knowledge Discovery and Data Mining*., 2002, pp. 28-39.

[15] Erica Kolatch, "Clustering Algorithms for Spatial Databases: A Survey," Dept. of Computer Science, University of Maryland, 2001.

[16] Harvey J. Miller, *Geographic Data Mining and Knowledge Discovery*, 2nd ed., Crc Press, Ed., 2009.

[17] Pavel Berkhin, "Survey Of Clustering Data Mining Techniques," Accrue Software, Relatório Técnico 2002.

[18] M. A. Wong J. A. Hartigan, "A K-Means Clustering Algorithm," *Applied Statistics*, vol. 28, pp. 100--108, 1979.

[19] Raymond T. Ng and Jiawei Han, "CLARANS: A Method for Clustering Objects for Spatial Data Mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 5, pp. 1003-1016, 2002.

[20] George Karypis and Eui-Hong (Sam) Han and Vipin Kumar, "CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling," *Computer*, vol. 32, pp. 68-75, Agosto 1999.

[21] Martin Ester and Hans-Peter Kriegel and Joerg Sander and Xiaowei Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226-231.

[22] Derya and Kut, Alp Birant, "ST-DBSCAN: An algorithm for clustering spatial-temporal data," *Data Knowl. Eng.*, vol. 60, no. 1, pp. 208--221, 2007.

[23] B. and Bhattacharyya, D.K. Borah, "An improved sampling-based DBSCAN for large spatial databases," in *Intelligent Sensing and Information Processing, 2004. Proceedings of International Conference on*., 2004, pp. 92-96.

[24] Gholamhosein Sheikholeslami and Surojit Chatterjee and Aidong Zhang, "WaveCluster: A Wavelet Based Clustering Approach for Spatial," *VLDB J.*, vol. 8, pp. 289-304, 2000.

[25] (2009, Nov.) Handling Large Amounts of Markers in Google Maps. [Online]. http://www.svennerberg.com/2009/01/handling-large-amounts-of-markers-in-google-maps/

[26] (2009, Nov.) MarkerCluster. [Online]. http://gmaps-utility-library.googlecode.com/svn/trunk/markerclusterer/1.0/docs/reference.html

[27] (2009, Nov.) ClusterMarker. [Online]. http://googlemapsapi.martinpearman.co.uk/articles.php?cat_id=1

[28] Wannes Meert, "Clustering Maps," Katholieke Universiteit Leuven, Tese de Mestrado 2006.

[29] (2009, Nov.) Travellr. [Online]. http://travellr.com/

[30] (2009, Nov.) "Travellr: Behind the Scenes of our Region-Based Clusters". [Online]. http://googlegeodevelopers.blogspot.com/2009/06/travellr-behind-scenes-of-our-region.html

[31] Martin Ester, Hans-Peter Kriegel and Xiaowei Xu Jörg Sander, "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications," *Data Mining and Knowledge Discovery*, vol. 2, pp. 169-194, Junho 1998.

[32] D. and Samal, A.K. and Leen-Kiat Soh Joshi, "Density-based clustering of polygons," in *CIDM*., 2009, pp. 171-178.

[33] Deepti Joshi and Ashok Samal and Leen-Kiat Soh, "A dissimilarity function for clustering geospatial polygons," in *GIS*, Ouri Wolfson and Divyakant Agrawal and Chang-Tien Lu, Ed.: ACM, 2009, pp. 384-387.

[34] Janine Gisele Le Sann, "O PAPEL DA CARTOGRAFIA TEMÁTICA NAS PESQUISAS AMBIENTAIS," *Revista do Departamento de Geografia*, pp. 61-69, 2005.

[35] Jacques Bertin, *Semiologie graphique: Les diagrammes - Les réseaux - Les cartes*, 1st ed.: l'Ecole des Hautes Etudes en Sciences, 1967.

[36] Julien Pastor, "Conception D'une Légende Interactive Et Forable Pour Le Solap," Faculté de Foresterie Et de Géomatique - Université Laval, Québec, Tese de Mestrado 2004.

[37] M. J. Atallah, "A linear time algorithm for the Hausdorff distance between convex polygons," *Information Processing Letters*, no. 17, pp. 207-209, 1983.

[38] George F. Jenks, "The Data Model Concept in Statistical Mapping," *International Yearbook of Cartography 7*, pp. 186-190, 1967.

[39] Paul S. Heckbert, "Nice Numbers For Graph Labels," in *Graphics gems.*, 1993, pp. 61-63.

[40] Thea Chiesa Jennifer Blanke, "The Travel & Tourism Competitiveness Report," World Economic Forum, 2009.

[41] Elzbieta Malinowski and Esteban Zimányi, "Spatial Data Warehouses: Some Solutions and Unresolved Problems," in *Proceedings of the 3 IEEE International Workshop*., 2007, pp. 1-6.

[42] E. Malinowski and E. Zimányi, "Spatial hierarchies and topological relationships in the spatial MultiDimER model," pp. 17-28, Julho 2005.

[43] Yvan Bédard, Marie-Josée Proulx, Martin Nadeau, Frederic Hubert and Julien Pastor Sonia Rivest, "SOLAP technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 1, pp. 17-33, Dec. 2005.

[44] X. Zhou, D. Truffet, and J. Han, "Efficient Polygon Amalgamation Methods for Spatial OLAP and Spatial Data Mining," 1999.

[45] R. Kothuri, A. Godfrind, and E. Beinat, *Pro Oracle Spatial for Oracle Database 11g*.: Apress, 2007.

[46] Chuck Murray, "Oracle Fusion Middleware User's Guide for Oracle MapViewer, 11g Release 1," 2009.

Anexo

A.1 Modelo de Estilos

No caso de **um objecto espacial** corresponder à forma geométrica **linha,** o modelo de estilos com uma coluna numérica é o seguinte:

- **Número de Colunas Alfanuméricas = 0:**

  As variáveis visuais possíveis para interpretar a coluna numérica são: *Tamanho*, *Luminosidade* e *Cor*. O outro estilo possível é a utilização do gráfico de barras.

- **Número de Colunas Alfanuméricas = 1:**

  - **Tipo de Dados = Nominal:**

    Para este contexto apenas se considera a variável *Cor* para interpretar a coluna alfanumérica. Para a coluna numérica as variáveis visuais possíveis são: *Tamanho* e *Luminosidade*. Outra possibilidade é combinar a variável *Cor* (coluna alfanumérica) com o gráfico de barras (coluna numérica). Por combinar, entenda-se, tanto aplicar a variável sobre o gráfico ou aplicar a cor sobre a linha e utilizar o gráfico de forma normal.

  - **Tipo de Dados = Ordinal:**

    Este contexto é semelhante ao anterior, mas em vez de se utilizar a variável *Cor* para mapear a coluna alfanumérica deve-se utilizar a variável *Tamanho* ou *Luminosidade*. A *Cor* não pode ser utilizada em simultâneo com a *Luminosidade*.

- **Número de Colunas Alfanuméricas $\geq$ 2 e $\leq$ 3:**

  Para este contexto existe a possibilidade de dois tipos de estilos compostos. Um que é composto por apenas estilos simples e outro que combina as variáveis visuais com gráficos. Existe relativamente uma alguma variedade de combinações. No entanto, em qualquer combinação, as restrições referidas nas tabelas Tabela 5 e Tabela 6 devem ser mantidas.

Na Tabela 5 verifica-se que a definição das primitivas espaciais, em função do tipo de dados e das variáveis visuais é semelhante para a forma espacial ponto e linha. Em consequência, o modelo de estilos para a forma geométrica linha torna-se semelhante ao do ponto. Contudo, é necessário alguns aspectos importantes na definição dos estilos possíveis.

Em primeiro lugar, não é abordado a utilização da textura ou da forma para representar tipo de dados nominais. Embora ambas as propriedades contenham um significado selectivo, quando aplicadas à forma geométrica linha considera-se que não permita uma leitura tão imediata do mapa.

Anteriormente, os estilos indicados para os *pontos* estão associados a uma coordenada latitude e longitude que reflecte a localização geográfica da entidade. Ora qualquer que seja o estilo associado a um ponto, este será representado sob a forma de um marcador com as devidas propriedades visuais associadas. No entanto, o mesmo não se verifica para as linhas quando os estilos se tratam de gráficos. Aliás, se um *Gráfico* associado a uma linha se tornasse um "substituto" do próprio objecto espacial, o mapa muito possivelmente tornar-se-ia confuso. Então, sempre que um estilo do tipo *Gráfico* estiver associado a uma linha, este não estará representado sob a forma de um marcador, ocultando a linha, mas sim sobre a linha, permitindo ao utilizador estabelecer a ligação de pertença do objecto espacial e o respectivo gráfico.

Á semelhança do modelo de estilos para os pontos, faz igualmente sentido um modelo de estilos para as linhas considerando duas colunas numéricas:

- **Número de Colunas Alfanuméricas = 0:**

  Com duas colunas numéricas, estas podem ser interpretadas pelas três variáveis visuais *Tamanho*, *Cor* e *Luminosidade*. Outra possibilidade é a utilização de um *Gráfico* (de barras ou circular). Adicionalmente a variável *Cor* e *Luminosidade* não podem ser utilizadas em simultâneo.

- **Número de Colunas Alfanuméricas = 1:**

  - **Tipo de Dados = Nominal:**

    Para este contexto faz sentido a utilização de um estilo composto que combina a variável *cor* com o gráfico de barras ou circular. Neste estilo o gráfico circular ou barras mapeia as duas colunas numéricas e a cor interpreta a coluna alfanumérica. Um outro tipo de estilo é um estilo que combina apenas variáveis visuais, não esquecendo as devidas restrições referidas para o modelo de estilos da linha nos contextos anteriores.

  - **Tipo de Dados = Ordinal:**

    Se no contexto anterior era utilizado a *cor* combinada com o gráfico de barras ou circular, pelo facto da coluna numérica ser nominal, neste caso a abordagem é semelhante, mas em vez de se aplicar a variável cor, pode-se aplicar a variável *Tamanho,* Cor ou *Luminosidade*. No entanto a *Luminosidade*

não pode ser aplicada sobre o gráfico circular (no entanto pode ser aplicada sobre a linha).

- **Número de Colunas Alfanuméricas ≥ 2 e ≤ 3:**

  Quando temos mais do que duas colunas alfanuméricas a utilização de estilos que combinem apenas variáveis visuais não é possível. Por exemplo, na presença de duas colunas alfanuméricas é possível utilizar o *Gráfico Barras*. Se uma das colunas for do tipo nominal e outra ordinal combina-se a aplicação das variáveis *Tamanho* e *Cor* com o gráfico. De notar que as variáveis *Cor* e *Luminosidade* só podem ser utilizadas em simultâneo se interpretarem tipo de dados alfanuméricos diferentes.

Outra particularidade relativamente às linhas é a possibilidade de mapear dados ordinais e quantitativos utilizando a variável *cor*. Por conseguinte, poderia surgir estilos compostos com a variável *cor* e *luminosidade* em simultâneo a mapear, cada uma delas, dados quantitativos ou ordinais. Neste tipo de situações, a utilização em simultâneo destas variáveis entraria em conflito. Portanto, é introduzida a restrição que, para qualquer *Estilo Composto* que interprete unicamente dados quantitativos (potencialmente com dados ordinais), não pode verificar a utilização simultânea de ambas as variáveis.

Através de um modelo de estilos definido para formas geométricas *linhas*, torna possível ao utilizador analisar visualmente novos conjuntos de dados comparativamente ao modelo de estilos definido por Ruben Jorge [**1**]. Um bom exemplo da sua aplicabilidade seria a análise da sinistralidade rodoviária em Portugal. Se estivermos interessados em analisar o número de acidentes e o número de vítimas (que engloba vítima mortal, ferido grave, ferido leve) com os dados agregados por estrada, então seria adequado utilizar o *Estilo Composto (Tamanho, Luminosidade)*. A partir deste estilo facilmente era possível concluir quais as estradas com mais acidentes, com mais vítimas, se um número elevado de acidentes implica um número elevado de vítimas mortais, entre outras análises. Se porventura quisermos comparar os dados anteriores por o tipo de via (Auto-estrada, estrada nacional, etc.) poderíamos utilizar o *Estilo Composto (Tamanho, Gráfico de Barras)*. Como o tipo de dados do atributo *tipo de via* é ordinal (arruamento, estrada municipal, IP/IC, estrada nacional, auto-estrada) a variável *tamanho* seria apropriada. Deste modo, era possível descobrir possíveis relações entre o tipo de estrada com o número de acidentes e número de vítimas.

Por fim, quando surgem casos em que o número de colunas numéricas é superior a dois, os estilos possíveis são os seguintes:

- **Número de Colunas Alfanuméricas = 0:**

  Para este contexto os estilos possíveis são apenas do tipo *Gráfico*.

- **Número de Colunas Alfanuméricas = 1:**

  O tipo de estilo possível para este contexto foi já apresentado anteriormente, quando se introduz o modelo de estilos para duas colunas numéricas. Esse corresponde ao estilo que combina a variável *Cor* com o tipo *Gráfico*.

- **Número de Colunas Alfanuméricas ≥ 2 e ≤ 3:**

  Neste contexto os estilos possíveis correspondem apenas àqueles que combinam variáveis visuais (para interpretar as colunas alfanuméricas) com o tipo *Gráfico* (que apresenta as colunas numéricas).

Nas situações em que estamos perante mais do que duas colunas numéricas o número de estilos torna-se um pouco limitado. Primeiro, como discutido anteriormente, a variável *textura* e *forma* não são adequadas quando aplicadas às linhas e, segundo, a variável *luminosidade* e *cor* entram em conflito quando ambas estão a interpretar dados contínuos.

Relativamente aos primeiros três primeiros pontos os estilos já foram abordados anteriormente, apenas diferindo o modo de aplicação. Isto é, o estilo não corresponde ao próprio objecto que explicita a localização geográfica da entidade, como era verificado com o tipo geométrico *ponto*.

Quando estamos na presença de **um objecto espacial** e este corresponde ao objecto geográfico **polígono,** associado a apenas uma coluna numérica:

- **Número de Colunas Alfanuméricas = 0:**

  As variáveis visuais possíveis para interpretar a coluna numérica são: *Luminosidade* e *Cor*. O outro estilo possível é a utilização do gráfico de barras.

- **Número de Colunas Alfanuméricas = 1:**

  - **Tipo de Dados = Nominal:**

    Para este contexto apenas se considera a variável *Cor* ou *Textura* para interpretar a coluna alfanumérica. Para a coluna numérica as variáveis visuais possíveis são: *Cor* e *Luminosidade*. Outra possibilidade é combinar a variável *Cor* (coluna alfanumérica) com o gráfico de barras (coluna numérica) (a variável cor pode ser tanto usada sobre o gráfico como usada no próprio polígono).

- **Tipo de Dados = Ordinal:**

  Neste contexto, para mapear as duas colunas podem ser utilizadas as variáveis *Luminosidade* ou *Cor* (apenas se não forem utilizadas para representar a coluna numérica). Outra possibilidade é combinar a variável *Luminosidade* sobre o gráfico de barras.

- **Número de Colunas Alfanuméricas ≥ 2 e ≤ 3:**

  Para este contexto existe a possibilidade de dois tipos de estilos compostos. Um que é composto por apenas estilos simples e outro que combina as variáveis visuais com gráficos à semelhança do que já foi definido anteriormente para contextos semelhantes (mas para outros objectos espaciais). Na analogia tem que se ter conta as restrições das tabelas Tabela 5 e Tabela 6.

O modelo de estilos para os polígonos tem, logo à partida, duas restrições que correspondem à impossibilidade de utilizar as variáveis *tamanho* e *forma*. Em contrapartida, a aplicação da variável *textura* nos polígonos tem um efeito mais imediato no utilizador (comparativamente com as linhas), e como tal a sua utilização é benéfica para interpretar dados nominais.

Quanto aos estilos possíveis para os polígonos, estes são semelhantes aos estilos já propostos para as outras formas geométricas, sendo apenas diferente a colocação dos gráficos. Quando um estilo *Gráfico* é aplicado aos polígonos, estes devem estar contidos dentro do polígono para facilmente o utilizador estabelecer a relação de pertença.

Até ao momento, apenas foi definido o modelo que reflecte parte da árvore de decisão dos polígonos. Em casos em que se verificam duas ou mais colunas numéricas, o modelo de estilos é o seguinte:

- **Número de Colunas Alfanuméricas = 0:**

  Para este contexto os estilos possíveis são apenas do tipo *Gráfico*.

- **Número de Colunas Alfanuméricas = 1:**

  - **Tipo de Dados = Nominal:**

    Neste contexto pode ser combinada a utilização da variável *Textura* ou *Cor* representada no polígono (para a coluna alfanumérica) e um gráfico sobre o polígono para representar as colunas numéricas;

- **Tipo de Dados = Ordinal:**

  Neste contexto pode ser combinada a utilização da variável *Luminosidade* ou *Cor* representada no polígono (para a coluna alfanumérica) e um gráfico sobre o polígono para representar as colunas numéricas;

- **Número de Colunas Alfanuméricas ≥ 2 e ≤ 3:**

  Neste contexto os estilos possíveis correspondem apenas àqueles que combinam variáveis visuais (para interpretar as colunas alfanuméricas) com o tipo Gráfico (que apresenta as colunas numéricas).

O modelo de estilos para os polígonos é igual na presença de duas ou mais colunas métricas. A razão desta definição é devido ao conflito das variáveis *luminosidade* e *cor* quando ambas representam dados quantitativos. Este conflito implica a não utilização de ambas as variáveis em simultâneo, o que reduz os estilos possíveis para cenários onde se verificam duas colunas numéricas.

Até ao momento ainda não foi referida uma questão pertinente. Neste caso em concreto, quando se está na presença de colunas numéricas pode-se recorrer à utilização de gráficos. Porém, existem diversos tipos de gráficos (gráfico de barras, gráfico circular, gráfico de linha, entre outros) e não existe um gráfico que seja mais adequado que qualquer outro numa dada situação.

Como já foi referido em [**1**], o gráfico circular é mais apropriado em situações que as diferentes colunas numéricas contribuem para uma dada métrica. Por exemplo, recordando a análise da sinistralidade em Portugal, se estivesse analisar o número de vítimas mortais segundo o tipo de via com os dados agregados por distrito, seria interessante a utilização do gráfico circular. Facilmente o utilizador percebia de que forma cada tipo de estrada contribui para o número total de vítimas mortais, para cada distrito.

Já o gráfico de barras é mais indicado para mostrar as alterações de dados em diferentes períodos de tempo ou ilustrar as comparações entre itens. Para o exemplo anterior, seria indicado para analisar, para o mesmo nível de agregação, o número de vítimas mortais nos diferentes anos. O gráfico de linhas é ideal para ilustrar as tendências nos dados que ocorram ao longo do tempo. Poderia ser aplicado, ao exemplo anterior, para ajudar o utilizador a perceber se a tendência do número de vítimas mortais para cada distrito é para aumentar ou para diminuir.

Por último, podem surgir casos com **dois objectos espaciais**. Para o modelo de estilos na presença de dois objectos espaciais é indiferente se estes sejam pontos ou polígonos, pois no caso dos polígonos são utilizados os centróides para as extremidades do arco (secção 3.2). Deste modo, o modelo de estilos para estes casos em concreto é o seguinte:

- **Número de Colunas Numéricas:**
  - **= 1:**

    Para interpretar uma coluna numérica pode ser utilizada a variável *Cor, Luminosidade* ou o gráfico de barras.

  - **> 2:**

    Para mapear as colunas numéricas, tipicamente, o estilo pode consistir num *Gráfico.* No entanto, se se estiver na presença de apenas duas colunas numéricas podemos combinar a variável *Cor* ou *Luminosidade* com o variável *Tamanho*

Composto com:

- **Tipo de objecto Espacial (Extremidade) e número de colunas alfanuméricas maior ou igual a um:**
  - **Ponto:**

    Ao estilo anterior pode compor o estilo com estilos para as extremidades. Cada variável mapeia uma coluna alfanumérica. No caso de a "coluna" ser do tipo ordinal então pode-se usar a variável *Tamanho* ou *Luminosidade*. Caso contrário pode-se utilizar a *Forma*, *Textura* ou *Cor*. Recordo, que não se pode utilizar mais do que três variáveis visuais em simultâneo.

  - **Linha:**

    Semelhante ao anterior. Cada coluna mapeia uma coluna alfanumérica. No caso de a "coluna" ser do tipo ordinal então pode-se usar a variável *Tamanho, Luminosidade* ou *Cor*. Caso contrário pode-se utilizar a *Cor*. Cada variável só pode ser utilizada uma vez.

O modelo de estilos com análises com dois atributos espaciais difere um pouco dos modelos de estilos anteriores. Sempre que o utilizador esteja em interacções com dois atributos espaciais em simultâneo a relação entre os objectos espaciais, em geral, será apresentada no seguinte formato (Figura 89):

Figura 89 - Formato visual que estabelece a relação espacial entre dois objectos.

A relação entre os dois objectos é realizada através do arco. Como as colunas numéricas correspondem aos dados que descrevem a relação entre os dois objectos espaciais, então o estilo que interpretará esses dados é associado ao arco.

O mesmo não se verifica para as colunas alfanuméricas. Cada coluna alfanumérica estará associada a um dos atributos espaciais. Logo, o estilo que mapear a coluna alfanumérica apenas se reflecte na extremidade correspondente.

Por fim, apesar de os estilos serem aplicados a objectos espaciais diferentes, deve-se manter a restrição de que não se pode utilizar a mesma variável visual mais do que uma vez.